DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/96M-17

AN INVESTIGATION OF PRELIMINARY FEATURE SCREENING USING SIGNAL-TO-NOISE RATIOS

THESIS
David B. Sumrell
Captain, USAF

AFIT/GOR/ENS/96M-17

Approved for public release; distribution unlimited

AFIT/GOR/ENS/96M-17

AN INVESTIGATION OF PRELIMINARY FEATURE SCREENING USING SIGNAL-TO-NOISE RATIOS

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

David B. Sumrell, B.S., M.S.

Captain, USAF

March 1996

THESIS APPROVAL

STUDENT: David B. Sumrell, USAF CLASS: GOR 96M-17

THESIS TITLE: AN INVESTIGATION OF PRELIMINARY FEATURE SCREENING USING SIGNAL-TO-NOISE RATIOS

DEFENSE DATE: 28 February 96

COMMITTEE:

Name/Title/Department	Signature
Advisor: Kenneth W. Bauer, Lt Col, USAF Associate Professor of Operations Research Department of Operational Sciences	Kenneth W Bauer
Reader: Daniel E. Reynolds Assistant Professor of Computing Science Department of Mathematics and Statistics	Daniel E. Reynolds

Preface

The purpose of this research effort was to explore the usefulness of a new saliency metric in a new saliency screening method. The new metric, the SN saliency metric, is based upon signal-to-noise ratios and incorporates the neural network's treatment of a reference noise feature into the calculation.

The SN saliency metric, in turn, was used in a new saliency screening method, the SN saliency screening method, which attempts to identify salient features in one screening run. This method actually interrupts the screening run at certain intervals, determines the least salient feature, removes that feature and continues the screening run. A feature's saliency is indicated by the order in which it is removed and the classification error rate's reaction to its removal.

No statistical validation of this saliency screening method is developed, but confidence in the method is bolstered empirically through its application to two example problems: Fisher's Iris Classification problem and the XOR problem. A designed experiment explores the consistency of the SN saliency metric across a range of neural network architectures. The SN saliency screening method is then applied to several neural network designs to assess the method's robustness both within a given neural network and across a range of neural network designs.

The inspiration behind this effort came primarily from three people: Lt Col Kenneth W. Bauer, who conceived the idea of the SN saliency metric and its potential application in an 'on the fly' saliency screening method; Professor Daniel E. Reynolds, who provided direction and invaluable feedback from a perspective not centered upon neural networks; and my wife, Dawn, who shouldered the weight of this effort along with me, who never lost confidence when mine faltered, and who, with eyes fixed upon the goal while mine were simply fixed, supported, encouraged and inspired me through to the very last step. The completion of this project is primarily a consequence of their faith in my ability.

David B. Sumrell

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

AFIT/GOR/ENS/96M-17

Abstract

A new saliency metric and a new saliency screening method are developed. This new metric, the SN saliency metric, is based upon signal-to-noise ratios, where the signal is provided by a sum of squared weights associated with a given feature, and the noise is based upon a sum of squared weights associated with a reference noise feature which is injected into the data. The resultant metric allows for a direct comparison of the feature of interest with a reference noise feature which is known to be nonsalient.

The SN saliency screening method, which uses the SN saliency metric, offers the potential of identifying salient features in one saliency screening run and is envisioned as an economical rough screening tool to be used prior to more refined screening efforts or more exhaustive training efforts. During the screening run, features are removed individually based upon their rank as determined by the SN saliency metric. The classification error rate's reaction to a given feature's removal helps confirm that feature's saliency.

Efforts at empirical validation of the new metric and saliency screening method center on application to two example problems: Fisher's Iris Classification problem and the XOR problem. Both examples use a feed-forward, fully-connected perceptron with one hidden layer and weight updates provided through back-propagation. These applications suggest that both the SN saliency metric and the SN saliency screening method produce consistent results across a range of neural network designs.

An Investigation of Preliminary Feature Screening Using Signal-to-Noise Ratios

I. Introduction

General Issue

The term GIGO, or *garbage in garbage out*, is familiar to many computer users. It suggests a dependence between the quality of a computer's output and the quality of the input. A similar dependence exists for classifiers between the quality of the input to the classifier and the accuracy of the resultant classification. In this context, quality of the input refers to that input's relevance to the classification problem, or to the predictive power inherent in the input. For classifiers, the term *saliency* refers to the strength or presence of this relevance or predictive power.

Identifying the salient input, or *saliency screening*, therefore, is an important step in building an accurate classifier. Accurate classification is not the only desirable result of successful saliency screening. The salient input will most likely be a smaller subset of the original input, thus reducing the amount of the input required to be collected and stored. Furthermore, reducing the amount of available input to the salient input reduces the amount of data required for training the artificial neural network. Finally, the classification accuracy may increase as nonsalient input, or *noise*, is eliminated from the training input.

Saliency screening requires both a saliency metric, i.e. a way to measure saliency, and a screening method based upon the saliency metric. Two current salient metrics are Ruck's saliency [7] and Tarr's saliency [9], each of which are described in more detail in Chapter 2. This study introduces a third saliency metric, the SN Saliency metric, which is based upon signal-to-noise ratios. This new metric is described in detail in Chapter 3.

Each of these three saliency metrics can be used by two saliency screening methods, the Belue-Bauer method [2] and the Steppe-Bauer method [8], each of which is described in more detail in Chapter 2. Each of these methods uses multiple training runs (30 and 10 recommended, respectively) of the neural network prior to assessing saliency and removing any nonsalient input from the original input. A third saliency screening method, introduced here as the SN Saliency Screening method, uses the signal-to-noise ratio saliency metric to assess saliency. Although not yet statistically validated, the potential for successful identification of salient variables in one training run enjoys empirical support. The economy of this third method is evident in the reduction of the recommended number of training runs with the original input set. The magnitude of the savings, primarily in run time, depends upon the size of the original input set and the particular neural network architecture.

If SN saliency screening is accomplished in only one run, the robustness of this method is critical. Since both the Belue-Bauer method and Steppe-Bauer method use multiple independent runs, they can calculate average saliency measures which should prove more resistant to the effects of random deviations. How robust is the SN Saliency Screening method in the face of these same random deviations?

A second consideration becomes the effect of neural network architecture on the value of the SN saliency metric. Does the success of the SN Saliency Screening method depend upon the structure of the neural network?

Confidence in this new method relies heavily upon the answer to those two questions. Chapter 3 describes the approach used to determine the robustness of this method both across independent training runs within a given neural network architecture, and across independent training runs across a range of neural network architectures. A finding that this method is robust in these two areas helps establish its credibility.

Due to time constraints, no direct comparison between the three saliency screening methods was conducted. This direct comparison among the three methods could be the subject of future research. Furthermore, the SN saliency screening method is not envisioned as a replacement for either the Belue-Bauer method or the Steppe-Bauer method. Each of these two saliency screening methods provides a finer, more rigorous screening tool due to the reliance upon multiple runs and established statistical testing. The SN saliency screening method, on the other hand, provides a less rigorous, though empirically satisfying screening tool. The author envisions a sequential use of the SN saliency screening method and either of the remaining two methods. The SN method provides an economical, rough initial screening of the original input set in order to eliminate the obvious noise, especially useful for rapid saliency screening for time critical applications, while the subsequent use of either the Belue-Bauer method or the Steppe-Bauer method provides a finer screening of the remaining input set in order to assess the saliency of any questionable or borderline input. The advantage of this sequential operation is that the original input set, i.e. the largest input set, is only used during one run, while the multiple runs required by the secondary screening use a subset of the original input set, a subset which has hopefully been significantly reduced in size. This reduction in size speeds training in the secondary screening by allowing for a smaller neural network. Additionally, since the reduced input set used during secondary screening contains less noise than the original input set, the resultant classifier may be more accurate.

II. Literature Review

This chapter covers definitions of terms commonly used when discussing neural networks, a summary of multilayer perceptrons, a brief description of the backpropagation training method, introductions to two saliency metrics (Ruck's saliency and Tarr's saliency) as well as discussions of two saliency screening methods (Belue-Bauer method and Steppe-Bauer) which use these saliency metrics. The two saliency metrics and two saliency screening methods are offered as examples of the current state of the saliency screening art.

Definitions

Back-propagation. A learning algorithm for updating weights in a multilayer, feed-forward, mapping neural network that minimizes mean squared mapping error. [3]

Classifier. The decision-making system built by the neural network. In a sense, the final set of weights. [7:7]

Epoch. A complete presentation of the data set being used to train the multilayer perceptron; also called a training cycle. [7:7]

Exemplar. The input data to a neural network is a finite set of solved cases. Each case is known as an exemplar or input vector. [7:7]

Feature. The individual measurements found in exemplars which contain information useful for distinguishing the various classes. In other fields, features are known as attributes or independent variables. [7:7]

Feedforward. Characterized by multilayer neural networks whose connections exclusively feed inputs from lower to higher layers; in contrast to a feedback network, or a recursive network, a feedforward network operates only until its inputs propagate to its output layer. An example of a feedforward neural network is the mulitlayer perceptron. [3]

Hidden Units. Those processing elements in multilayer neural network architectures which are neither the input layer nor the output layer, but are located in between these and allow the network to undertake more complex problem solving. [3]

Learning Algorithms. In neural networks, the equations which modify some of the weights of processing elements in response to input and output values. [3]

Multilayer Perceptron. A multilayer feedforward network that is fully connected (each node in any given layer is connected to every node in the next layer) and which is typically trained by the back-propagation learning algorithm. [3]

Neural Network. An information processing system which operates on inputs to extract information and produces outputs corresponding to the extracted information. [3]

Single-layer Perceptron. A type of neural network algorithm used in pattern classification problems and trained with supervision. Connection weights and thresholds in a perceptron can be fixed or adapted using a number of different algorithms. [3]

Supervised Training. A means of training adaptive neural networks which requires labeled training data and an external teacher. The teacher knows the correct response and provides an error signal when an error is made by the network. [3]

Weight. A processing element (or neuron or unit) need not treat all inputs uniformly. Processing elements receive inputs by means of interconnects (also called 'connections' or 'links'); each of these connections has an associated weight which signifies its strength. The weights are combined to calculate the activations. [3]

Multilayer Perceptron

Figure 1 shows a single-output perceptron. The perceptron receives a weighted sum of the M features and the bias term. The perceptron then performs a mathematical transformation on this weighted sum, and this transformation serves as the perceptron's output. The superscripts associated with the weights and the output are not powers; rather, they identify the layer with which the weight or output is associated. For example,

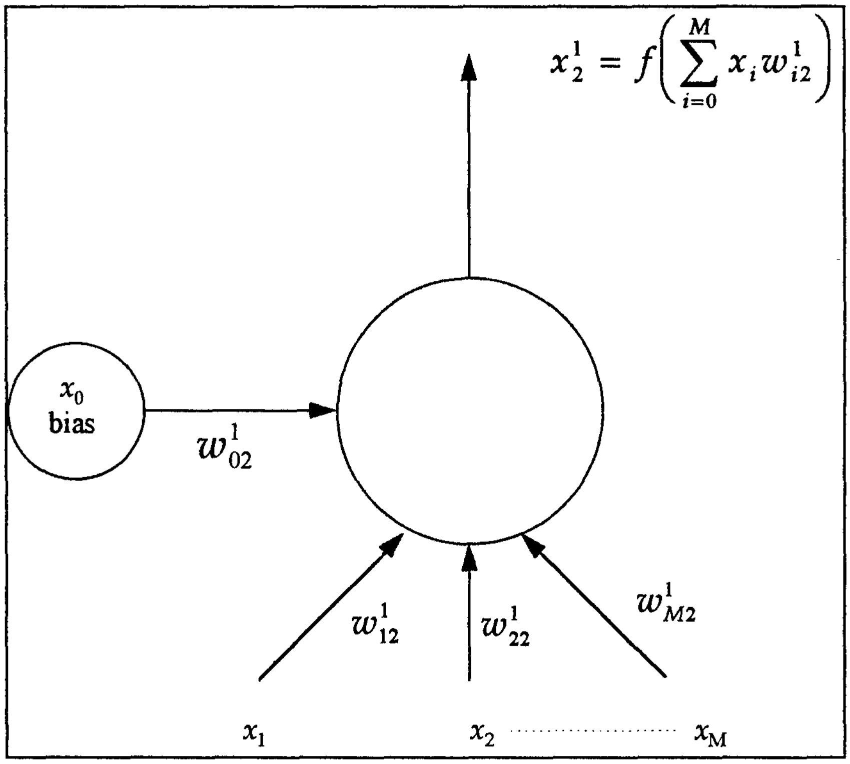

Figure 1. Single-Output Perceptron

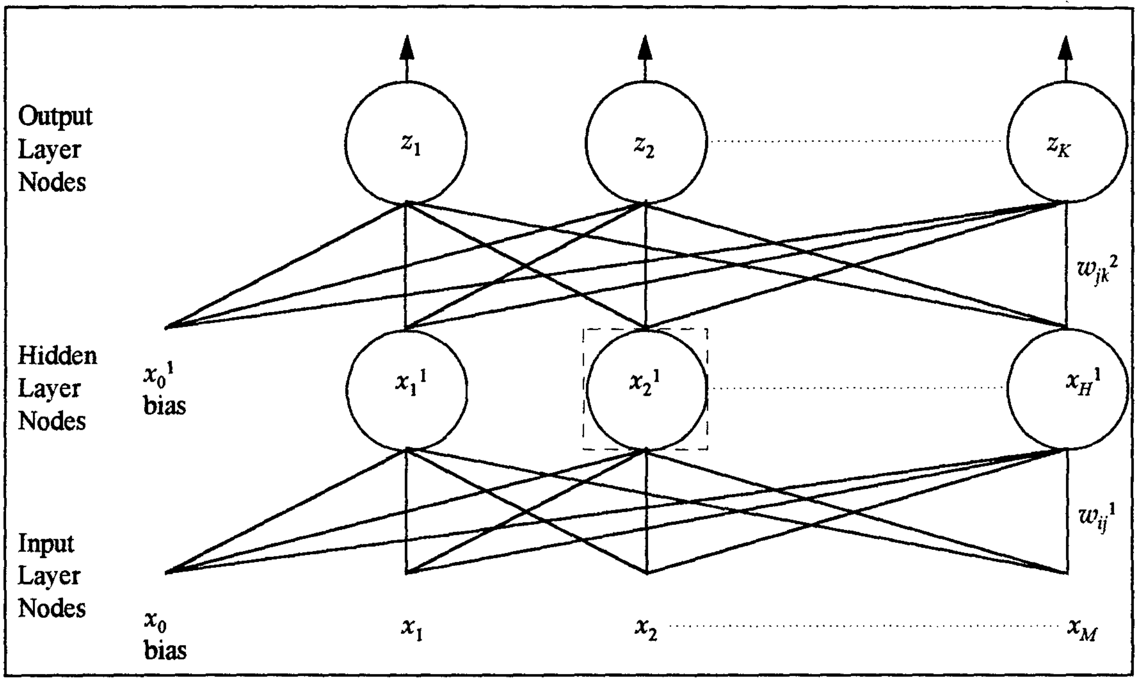

Figure 2. Multilayer Network

the superscript '1' associated with the weights indicates that the weights connect the first layer with its previous layer (by convention, the input layer is considered the 0 layer). When associated with the perceptron's output, the superscript '1' indicates that the perceptron is located in the first layer.

Figure 2 shows a feedforward, fully-connected multilayer perceptron with one hidden, or middle, layer. Each node in the input layer receives its input from a specific feature in the exemplar. Figure 2 shows M input nodes corresponding to M features. The bias term is provided by the neural network, and its value is always 1. Each input node, including the bias node, is connected to every hidden, or middle, node. In Figure 2 there are H hidden nodes. The bias node in the middle layer is similar to that in the input layer, and its value is also set to 1.

Each of the H middle nodes receives a weighted sum of the values associated with the M features included in the current exemplar:

$$x_j^1 = \sum_i w_{ij}^1 x_i \tag{1}$$

where x_j^1 is the input for the *j*-th node in the middle layer, w_{ij}^1 is the weight connecting the *i*-th input node with the *j*-the middle node and x_i is the input from the *i*-th input node.

The actual output of each of the hidden nodes and each of the output nodes is some transformation of the weighted sum shown in Equation (1). Although any of a number of transformations may be used, this study uses the sigmoid function to transform the weighted sums. Figure 3 provides a graphical representation of the sigmoid function while Equation (2) shows the calculations associated with the outputs from the hidden and output layers.

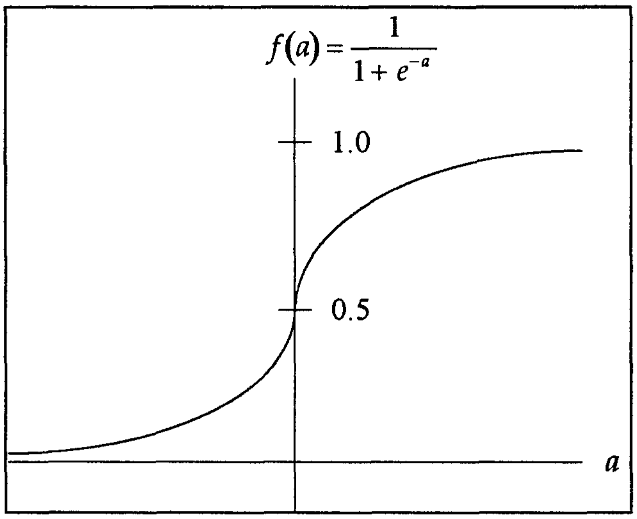

Figure 3. The Sigmoid Function

$$z_j^1 = \frac{1}{1+\exp\left(\sum_i w_{ij}^1 x_i\right)} \qquad z_k^2 = \frac{1}{1+\exp\left(\sum_j w_{jk}^2 z_j^1\right)} \tag{2}$$

In Equation (2), z_j^1 is the output of the *j*-th hidden node, w_{ij}^1 is the weight connecting the *i*-th input node with the *j*-th hidden node, x_i is the value of the *i*-th feature, z_k^2 is the output of the *k*-th output node and w_{jk}^2 is the weight connecting the *j*-th hidden node to the *k*-th output node.

The backpropagation training algorithm updates these weights, either after each exemplar (instantaneous updating) or after each epoch (batch updating). For the multilayer network shown in Figure 2, the weights between the input layer and the hidden layer update in the following manner:

$$w_{ij}^1(t+1) = w_{ij}^1(t) + \eta\delta_j^1 x_i + \alpha\left[w_{ij}^1(t) - w_{ij}^1(t-1)\right] \tag{3}$$

where *i* is associated with the *i*-th input node, *j* is associated with the *j*-th hidden node, $w_{ij}^1(t+1)$ refers to the updated weights, $w_{ij}^1(t)$ refers to the current weights and $w_{ij}^1(t-1)$ refers to the previous weights. The term δ_j^1 designates the error derivative of the *j*-th hidden unit. Assuming that each hidden unit output is a sigmoid transformation of its sum of weighted inputs, δ_j^1 is calculated by Equation (4):

$$\delta_j^1 = z_j^1\left(1 - z_j^1\right)\left(\sum_k \delta_k^2 w_{jk}^2\right) \tag{4}$$

where z_j^1 is the j-th hidden node's output, δ_k^2 is the error derivative for the k-th node in the output layer (associated calculation shown by Equation (6)), and w_{jk}^2 is the weight connecting the j-th hidden node with the k-th output node.

The weights connecting the hidden and output layers update in a similar manner. Equation (5) shows the actual weight updates:

$$w_{jk}^2(t+1) = w_{jk}^2(t) + \eta\delta_k^2 x_i + \alpha\left[w_{jk}^2(t) - w_{jk}^2(t-1)\right] \tag{5}$$

where j refers to the j-th hidden node, k refers to the k-th output node, δ_k^2 is the error derivative of the k-th output node, and the sequence of updated, current and previous weights is analogous to the sequence described for Equation (3). Equation (6) shows the calculation of the error derivative of the k-th output node:

$$\delta_k^2 = z_k^2\left(1 - z_k^2\right)\left(d_k - z_k^2\right) \tag{6}$$

where z_k^2 is the actual output for the k-th output node and d_k is the desired output of the k-th output node.

In Equations (3) and (5), the terms η and α refer to the learning rate step size and the momentum rate, respectively. The learning rate step size moderates the amount of the error derivative which will be included in the updated weight, while the momentum rate determines how much the updated weight will depend upon the magnitude and direction of previous weights. In a way, the learning rate step size determines how large a step the updated weight will take in a given direction, while the momentum rate determines how much the direction will change.

Saliency Metrics

This section describes two saliency metrics: Ruck's saliency and Tarr's saliency. Both of these saliency metrics provide an ordering of the given features, based upon the value of the metric calculation, which indicates relative saliency among the features.

Ruck's Saliency. Ruck's metric measures the saliency of a given feature by summing the partial derivatives of the network outputs with respect to that given feature. The formula is shown below:

$$\Lambda_i = \sum_P \sum_M \sum_R \sum_K \left| \frac{\partial z_k^2}{\partial x_i} \left(\vec{\mathbf{x}}_p^{m(r)}, \vec{\mathbf{w}} \right) \right| \tag{7}$$

P is the number of exemplars, M is the number of features, R is the number of steps that the range of each feature is uniformly divided into, and K is the number of network outputs. The vector with which the partial derivative is calculated is the *p*-th exemplar with its *m*-th feature replaced by the value associated with that feature's *r*-th step. The weights are the final estimates produced by the trained network.

This saliency metric orders the features from most salient to least salient based upon the features' metric values. Higher values indicate higher relative saliency, while lower values indicate lower relative saliency.

Tarr's Saliency. Tarr's metric measures the saliency of a given feature by summing the squared values of the weights connecting that feature's input node to the middle nodes. The calculation is shown below:

$$\tau_i = \sum_{m=1}^{M} \left(w_{im}^1 \right)^2 \tag{8}$$

The following discussion of weight updates contains the philosophy behind this metric:

> When a weigh is updated, the network moves the weight a small amount based on the error. Given that a particular feature is relevant to the problem solution, the weight would be moved in a constant direction until a solution with no error is reached. If the error term is consistent, the direction of the movement of the weight vector, which forms a hyper-plane decision boundary, will also be consistent. . . .If the error term is not consistent, which can be the case on a single feature out of the input vector, the movement of the weight attached to the node will also be inconsistent. In a similar fashion, if the feature did not contribute to a solution, the weight updates would be random. In other words, useful features would cause the weights to grow, while weights attached to non-salient features would simply fluctuate around zero. [10:44]

As with Ruck's saliency, large values of this metric indicate high relative saliency, while small values indicate low relative saliency.

Saliency Screening Methods

This section describes two saliency screening methods: the Belue-Bauer screening method and the Steppe-Bauer screening method. Both of these methods inject a known noise feature into the original data set, then use the saliency measure of the injected noise feature as a baseline against which the saliency of the features of interest is determined.

Belue-Bauer Saliency Screening. Recognizing that the then current method of saliency screening retained features subjectively based upon relative saliency metrics, Belue and Bauer set out "to develop a method which takes into consideration the saliency of a feature relative to the saliency of a known irrelevant feature." [1:115] The resultant method is summarized below:

(1) Introduce a noise feature to the original set of feature vectors.
(2) Train the network.
(3) Compute the saliency of all features (using either Ruck's Saliency or Tarr's Saliency).

(4) Repeat steps 2 and 3 at least 30 times (with weights being randomly initialized and training and test sets being randomly selected at the beginning of each training cycle).
(5) Assume the average saliency of noise is normally distributed and find the upper one-sided (α X 100) percent confidence interval for the mean value of the saliency of noise.
(6) Choose only those features whose average saliency value falls outside this confidence interval.
(7) Retrain the network with the salient features. [1:115-116]

In their conclusions and recommendations, Belue and Bauer report that "the introduction of noise as a feature input provides a method for determining the significance of a set of features by comparing their saliency to the saliency of the injected noise." [1:119] After the nonsalient features were removed "the multilayer perceptron trained quicker and exhibited a lower output error and classification error." [1:119] Furthermore, a comparison of Ruck's saliency and Tarr's saliency showed that they "ordered the features similarly with the same conclusions reached even though the measurement scales were different." [1:119]

Steppe-Bauer Saliency Screening. Building upon the work begun by Belue and Bauer, Steppe and Bauer develop a "saliency screening procedure for identifying noisy features . . . based on statistically comparing the mean saliency of candidate features to the mean saliency of a noisy feature." [9:181] The procedure is summarized below:

(1) Augment feature set with a noise feature, x_n.
(2) Train neural net to minimize *training-test* set error.
All nets should ideally use a minimal network structure with no redundant middle nodes or features.
All nets should ideally converge to a local minimum and not a saddle point.
(3) Compute the feature saliency [modified Ruck's saliency]:

$$\hat{\Lambda}_i^{\text{data}} = P^{-1}\sum_{p=1}^{P}\sum_{k=1}^{K}\left|\frac{\partial z_k}{\partial x_i}(\mathbf{x}^p, \hat{\mathbf{w}})\right| \tag{9}$$

for each of the features, including x_n.

(4) Repeat steps (2) and (3) a minimum of ten times (N = 10), using random initialization of weight parameters and random data set partitioning.

(5) Select 'family' significance level, α.

(6) For each feature do an individual hypothesis test as follows [see Chapter 3: Paired *t*-Test]

(a) Compute $\overline{D}_i$ and $S^2_{\overline{D}_i}$

(b) Compute the test statistic t*

(c) Determine the Bonferroni critical value $B = t_{\frac{\alpha}{M'},\nu}$.

(d) Evaluate the test statistic as follows:

- If t* $\leq B$, the null hypothesis can not be rejected for feature i.

  Conclusion: feature i is nonsalient, since the difference between the i-th feature's saliency and the noise feature's saliency is not statistically different from zero at the α 'family' significance level.

- If t* > B, reject the null hypothesis for feature i.

  Conclusion: feature i is salient, since there is a statistical difference at the α 'family' significance level between the saliency of the i-th feature and the saliency of the noise feature.

(7) Eliminate the nonsalient features and retrain the network with only the salient features.

Steppe writes that "conservative results are common with this procedure. That is, a nonessential feature, having little or no bearing on the classification accuracy, may be identified as salient if it is statistically different from noise." [9:126] For this reason, Steppe recommends that "features with relatively low test statistics may warrant further consideration in the context of a feature selection process." [9:127]

III. Methodology

This chapter will introduce both the new saliency metric, based on signal-to-noise ratios, and a saliency screening method which uses this metric. In order to investigate the robustness of this saliency screening method both across a range of neural network architectures and across different runs at a particular architecture, an experimental design region will be developed. The results of the runs generated by this design region will be analyzed both statistically and graphically in an attempt to answer four research questions presented in this chapter. The specific statistical tests to be employed are also discussed.

Screening with Signal-to-Noise Ratios

A New Saliency Metric. The saliency metric proposed in this study resembles Tarr's saliency metric in that they both rely upon a sum of squared weights. However, this metric differs from both Tarr's saliency metric and Ruck's saliency metric in that the saliency metric for a given feature is actually a direct comparison of that feature to an injected noise feature.

Equation (10) shows the calculation of this saliency metric:

$$SN_i = 10 \log \frac{\sum_{j=1}^{J}\left(w_{ij}^{1}\right)^2}{\sum_{j=1}^{J}\left(w_{Nj}^{1}\right)^2} \tag{10}$$

where SN_i is the value of the saliency metric for the *i*-th feature, *J* is the number of hidden nodes, and w_{Nj}^{1} represents the set of weights connecting the injected noise feature, x_N, to the hidden nodes. The transformation of the ratio converts the saliency metric to a decibel scale.

The philosophy behind this metric is similar to that expressed by Tarr in Chapter 2. Salient features should produce larger weights and, therefore, larger sums of squared weights. The ratio of a salient feature's sum of squared weights to x_N's sum of squared weights should be significantly larger than one, and the final value of the metric should be significantly larger than zero.

A nonsalient feature, on the other hand, will likely generate weights that are closer in value to those generated by x_N, so the ratio of the respective sums of squared weights will be closer to one, perhaps less than one, and the subsequent transformation would produce an SN value close to zero, perhaps less than zero.

Salient features, therefore, will likely generate SN values significantly larger than zero, while nonsalient features will likely generate SN values not significantly larger than zero, and perhaps even less than zero. Furthermore, these SN values should rank the features in the same way that both Tarr's saliency and Ruck's saliency do. These anticipated characteristics are employed in the following saliency screening method.

SN Saliency Screening Method. The following saliency screening method uses the SN saliency metric to distinguish between salient and nonsalient features:

(1) Add a noise feature, x_N, to the original feature set.

(2) Begin training the neural network.

(3) Interrupt training after the saliency metric values have stabilized.

(4) Identify the feature with the lowest SN value and remove it from further training.

(5) Continue training the neural network.

(6) Repeat steps (3) - (5) until all of the features in the original set have been removed.

(7) Finish/discontinue training the neural network.

(8) Compare the reaction of the test set classification error rate to the removal of the individual features.

(9) Retain the first feature whose removal caused a significant increase in the test set classification error rate, as well as all features which were removed after that first salient feature.

This method depends heavily upon robust feature ranks provided by the SN saliency metric. If the rank proved to be inconsistent from one run to the next at a given neural network architecture, or if the ranks generated by different network architectures proved to be inconsistent, then this method would be unreliable at best. In order to investigate the issue of robust rank, this study exercised this method over an experimental design region, described in the next section.

Experimental Design

This section introduces the experimental design used by this study and the specific research questions used to guide this effort. Additionally, this section provides a brief description of the specific statistical tests and graphical analysis used to answer each question.

Experimental Design Region.

Table 1 summarizes the design region used in this study. The resultant full-factorial 3^3 design contains 27 different design points, each

Table 1. Experimental Design Region

		Level		
		Low	Middle	High
Factor	Number of Middle Nodes (N = Number of Features)	N	2N	3N
	Learning Rate Step Size	0.1	0.5	0.9
	Momentum Rate	0.1	0.5	0.9

corresponding to a different neural network architecture, across which the robustness of the feature rank provided by the SN metric will be investigated. The designed experiment includes ten independent runs, or replicates, at each design point, which will aid in determining feature rank robustness across independent runs at a given point. Since the focus of this designed experiment is feature rank robustness, the SN Saliency Screening method is not applied during these runs. Instead, each neural network trains for 2000 epochs, and the final set of trained weights provides the input for statistical and graphical analysis.

This experimental design holds other neural network architecture parameters constant or in a constant range. Table 2 summarizes these factors and their settings. Each of the 270 runs in the design region uses a different random number seed, which generates the random numbers used to randomly initialize the weights, randomly partition the data set into training and test sets, and randomly order the presentation of the training data. This use of independent random number seeds ensures that each of the 270 runs is an independent event. [9:112]

Table 2. Remaining Architecture Factors

Factor	Setting
Range of Weight Initialization	-0.5 to 0.5
Type of Learning Rate	Constant
Type of Data Normalization	Gaussian
Number of Epochs	2000

Research Questions. The experimental design described in the previous section aims to answer four specific questions. The associated statistical test for each question is described in the following section (Statistical Testing).

(1) Are the signal-to-noise ratio distributions identical for all features? If they are, this suggests that the SN metric is unable to distinguish between salient and nonsalient features. If the distributions are different, this metric is detecting some difference among the features, and this difference is most likely a difference in saliency.

Associated statistical test: Kruskal-Wallis H-Test for Comparing k Population Distributions

Additional analytical test: Graphical analysis of each feature's rank distribution

(2) If the distributions are different, how does each feature's distribution compare to that of x_N, the injected noise feature? Salient features should generate larger SN values than that generated by x_N. Therefore, salient features should generate a SN distribution

which is shifted to the right of the SN distribution for x_N. Nonsalient features, on the other hand, will not likely generate consistently larger SN values than that generated by x_N. As a result, nonsalient features will not likely generate a SN distribution which is shifted to the right of the SN distribution for x_N. An SN distribution which in not shifted to the right of the SN distribution for x_N suggests that the associated feature is nonsalient.

Associated statistical test: Paired *t*-Test

Additional analytical tools: 'Scree'-type plot of *t*-Test null hypothesis rejections by feature

(3) Are the feature ranks obtained from the signal-to-noise ratios consistent, both across differing neural network architectures and across repeat runs at a constant neural network architecture? This specifically addresses the robustness issue. If the ranks are not consistent, then the saliency screening method must be considered unreliable in those architectural design regions which produce the inconsistent ranks.

Associated statistical test: Spearman's Rank Correlation Test

(4) Does network architecture affect the results? The results of the 270 runs specified in the designed experiment may yield some general guidelines for which architectures to use and which to avoid when using this saliency screening method

Associated statistical test: Kruskal-Wallis H-Test for Comparing k Population Distributions, Paired *t*-Test, Spearman's Rank Correlation Test

Additional analytical tools: graphical analysis of factor effects on individual feature rank

Statistical Testing. Each of the research questions makes use of one or more of the following statistical tests.

Kruskal-Wallis H-Test for Comparing k Population Distributions. [5:697-700] This test compares the population distributions of the ranks of all of the features in the original data set and the injected noise feature. The ranks are ascending ranks, i.e. the feature with the lowest SN value receives the lowest rank, or '1', while the feature with the highest SN value receives the highest rank, or k, where k is the number of features. This assignment of an ascending rank applies across all statistical tests and graphical analyses which consider rank. A helpful memory aid is 'bigger is better.'

Furthermore, any ties are broken by assigning an average rank to each of the tied features. For example, if two features have the same SN value and tie for a rank of 15, then they each would receive a rank of 14.5 (the average of 15 and 14, the next lower rank). This tie-breaking method is also consistent across all of the statistical tests used in this study which use rank.

This test uses the following hypotheses:

H_0: The k population distributions are identical.
H_a: At least two of the population distributions differ in location.

As described in the previous section, this study hopes to reject H_0 and conclude that at least two of the population distribution differ in location. The test statistic calculated to test H_0 is given by Equation (11):

$$\text{Test Statistic: } H = \frac{12}{n(n+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(n+1) \qquad (11)$$

where

n_i = Number of measurements in sample from population i.
R_i = Rank sum for sample i, where the rank of each measurement if computed according to its relative size in the overall set of $n = n_1 + n_2 + \cdots$ n_k observations formed by combining the data from all k samples.

The test statistic value is compared to the following rejection region:

Reject H_0 if $H > \chi_\alpha^2$ with $(k - 1)$ degrees of freedom.

The test uses the following assumptions:

The k samples are randomly and independently drawn.
There are five or more measurements in each sample.

Unfortunately, our application does not meet these assumptions. The SN values within a given run are not independently drawn. Therefore, no inference can be drawn from the results of this test. However, with the violation of the assumption in mind, the results of this test might be used for a less rigorous purpose, such as a simple illustration.

The Paired *t*-Test. [4:312-315] This test is a special case of the two-sample *t*-test where the observations are collected in pairs. While it is assumed that conditions are homogeneous within pairs of observations, the conditions are allowed to vary between pairs of observations. The key to this test is the difference between the observations within each pair. If each observation within the pair comes from the same distribution, than the mean difference should approach zero. A mean difference which is significantly different from zero suggests that the observations do not come from the same, or identical, or even similar distributions.

This test will be run on the ten replicates associated with each design point, and on the 270 runs associated with the overall design region. This is a paired test, pairing the injected noise variable with each of the original features in turn. If there are k original features, this necessitates a 'family' of k paired t-tests at each design point and for the overall design region. According to Bonferroni, in order to infer conclusions based on the test result at a 'family' confidence level of α, the k individual tests included in the 'family' must be conducted at a confidence level of $\frac{\alpha}{k}$. [6:164-165]

The hypotheses used in this test are given below:

$$H_0\text{: } \mu_D = 0$$
$$H_a\text{: } \mu_D > 0$$

where $\mu_D = E(\ SN_i - SN_N) = E(SN_i) - E(SN_N)$ where SN_i is the signal-to-noise ratio for the i-th feature, and SN_N is the signal-to-noise ratio for the injected noise feature, x_N. The one-tailed test is used to detect a positive mean difference, which suggests that the SN distribution for the given feature is shifted to the right of the SN distribution for x_N , which indicates that the feature is more salient than x_N.

Equation (12) shows the test statistic:

$$t_0 = \frac{\overline{D}}{S_D/\sqrt{n}} \tag{12}$$

where the sample mean is given by:

$$\overline{D} = \frac{\sum_{j=1}^{n} D_j}{n} \tag{13}$$

the sample variance is given by:

$$S_D^2 = \frac{\sum_{j=1}^{n} D_j^2 - \left[\left(\sum_{j=1}^{n} D_j \right)^2 \Big/ n \right]}{n-1} \qquad (14)$$

and the difference is given by:

$$D_j = SN_{ij} - SN_{Nj} = SN_{ij} - 0 = SN_{ij} \qquad (15)$$

where D_j is the difference for the j-th run, SN_{ij} is the signal-to-noise ratio for the i-th feature for the j-th run, and SN_{Nj} is the signal-to-noise ratio for the injected noise feature for the j-th run. For this test, reject H_0: $\mu_D = 0$ (implying that $\mu_D > 0$) if $t_0 > t_{\alpha/k, n-1}$.

This test assumes that the observations are paired and that, within pairs, the observations are taken under homogeneous conditions, although this is not required between pairs. Assume that $SN_i \sim N(\mu_i, \sigma_i^2)$ and $SN_N \sim N(\mu_N, \sigma_N^2)$. In this case, SN_N is not normally distributed; in fact, it is a constant. However, this does not affect the assumption that the differences (D_j) are normally distributed.

Spearman's Rank Correlation Test. [5:715-719] This test aims to answer the third research question by taking a pair of runs and calculating the correlation between the ranks assigned for each given feature, using the hypotheses shown below:

> Null Hypothesis: H_0: There is no association between the rank pairs.
> Alternative Hypothesis: H_a: There is an association between rank pairs (a two-tailed test). Or H_a: The correlation between the rank pairs is positive (or negative) (a one-tailed test).

A positive correlation between the rank pairs suggests that the rank consistent across the tested space. Therefore, this study uses the one-tailed test. Equation (16) shows the test statistic:

$$r_s = \frac{n\sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{\sqrt{\left[n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2\right]\left[n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2\right]}} \qquad (16)$$

where x_i and y_i represent the ranks of the *i*-th pair of observations (in this case, the ranks assigned to the *i*-th feature).

The rejection region is described below:

> For a two-tailed test, reject H_0 if $r_s \geq r_0$ or $r_s \leq r_0$, where r_0 is the critical value of Spearman's rank correlation coefficient for the given sample size and confidence level ($\alpha/2$). For a one-tailed test, reject H_0 if $r_s \geq r_0$ (for an upper-tailed test) or $r_s \leq r_0$ (for a lower-tailed test), where r_0 is the critical value of Spearman's rank correlation coefficient for the given sample size and confidence level (α).

This study uses the rejection region associated with an upper-tailed test.

The Spearman Rank Correlation Test assumes that the paired observations have been randomly selected and are independent. As described in the Experimental Design Region section, the use of separate random number seeds for each run ensures independence between the runs and satisfies this test's independence assumptions.

Example Problems

The experimental design and statistical tests are run with two example problems: Fisher's iris data and the exclusive-or, or XOR, problem.

Fisher's Iris Classification Problem. This problem contains data from three distinct types of irises: iris setosa, iris versicolor, and iris virginica. The data measures four characteristics: sepal length, sepal width, petal length and petal width. During this study, these four characteristics are known as Variable 1, Variable 2, Variable 3 and Variable 4, respectively.

In order to add some nonsalient features to this data set, each individual characteristic is resorted randomly and reinserted into the data set as a corresponding noise variable. For example, the data associated with Variable 1, or sepal length is resorted randomly, and this new feature is labeled Noise 1 and reinserted into the original data set. This process expands the data set to eight features (Variables 1-4 and Noise 1-4).

The last feature added to the data set is the injected noise feature against which the saliency of the other eight features will be measured. This injected noise feature, known as Noise, is a uniformly-distributed random variable with a range of (0, 1). Thus, the final data set contains nine features.

Since there are three groups into which each iris may be classified, three binary classification variables are included in the data set. If the iris belongs to a given group, the associated classification variable takes the value '1'; otherwise, the classification variable has a value of '0'. This final addition boosts the data set to nine features and three classification variables.

XOR Problem. The XOR problem is most easily described by Figure 4. The two groups are determined by the values of Variable 1 and Variable 2. If the product of the two variables is positive, the object belongs to Group 1. If the product of the two variables is negative, the object belongs to Group 2.

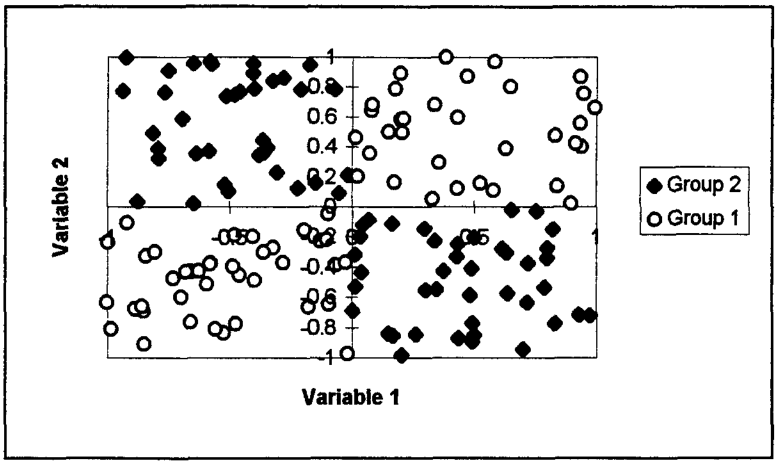

Figure 4. The XOR Problem

The data set consists of five features, each of which is a uniformly-distributed (-1,1) random variable. The first two features, Variable 1 and Variable 2, determine the object's correct classification, as described above. The remaining three features, Noise 1, Noise 2 and Noise 3, serve as the nonsalient features. A sixth uniformly-distributed (-1,1) random variable is inserted as the reference noise variable, Noise, against which the saliency of the original five features is measured. Finally, two binary classification variables, Group 1 and Group 2, are added. If the object belongs to Group 1, then the value of Group 1 is '1'; otherwise, the value of Group 1 is '0'. Group 2's values are determined in the same way. Thus, the final data set has six features (Variables 1 and 2, Noise 1 - 3, and the injected noise feature, Noise) as well as two classification variables (Group 1 and Group 2).

Exercising the SN Saliency Screening Method

After the experimental design is accomplished and associated analysis is completed, the SN Saliency Screening method is applied using both Fisher's Iris Classification problem and the XOR problem. The method is run ten times at a given neural network architecture and one time each at three different neural network architectures to allow an examination of the robustness of the method both within a specific architecture and across different architectures. The analysis focuses on differences in the subset of salient features retained by the method both among the ten replicates at a given neural network architecture and across the runs accomplished at the different neural network architectures. Retention of a consistent subset of salient features suggests robustness of the SN Saliency Screening Method.

IV. Results

This chapter is divided into two sections, corresponding to the designed experiment and the application of the SN Saliency Screening method. Each of these sections addresses both the Fisher Iris Classification problem and the XOR problem. The first section confirms the robustness of the features' ranks assigned by the SN metric within and across most neural network architectures. The second section assesses the robustness of the SN Saliency Screening method both within and across neural network architectures.

Experimental Design

This section presents the various calculations and analyses performed on the data obtained from the designed experiment using both the Fisher Iris Classification problem and the XOR problem. The results suggest that for both problems, the feature ranks assigned by the SN Saliency metric exhibit robustness within and across most neural network architectures. Certain architectures, especially those characterized by high momentum rates, produce inconsistent feature ranks, suggesting that high momentum rates should be avoided when applying the SN Saliency metric.

The results associated with the two problems are organized below according to the research questions and associated statistical testing and graphical analysis described in Chapter 3.

Fisher Iris Classification Problem

(1) Are the signal-to-noise ratio distributions identical for all features? The statistical test associated with this question is the Kruskal-Wallis H-Test for Comparing k Population Distributions. Table 3 begins the calculation of the test statistic by presenting the values of the SN Saliency metric at design point 9 - 0.1 - 0.1 (i.e. nine middle nodes, $\eta = 0.1$, $\alpha = 0.1$).

Table 3. SN Saliency Values for Design Point 9 - 0.1 - 0.1

Nodes	9	Learning Rate Step Size 0.1					Momentum Rate 0.1			
	Replicates									
	1	2	3	4	5	6	7	8	9	10
Noise	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Noise 1	4.98	-0.19	-0.17	-1.69	8.24	4.82	-1.21	-10.75	-8.64	-4.90
Noise 2	4.31	2.11	0.94	4.59	8.61	6.38	3.47	-0.93	-3.18	-2.52
Noise 3	-0.01	-3.76	-5.24	1.53	1.61	-3.16	-1.08	-1.70	-1.68	-2.42
Noise 4	4.01	1.38	-0.50	-3.32	9.08	6.10	-0.95	-3.45	-7.19	-3.82
Bias	18.69	11.42	11.64	13.38	18.99	17.15	12.22	10.26	11.53	9.43
Variable 1	5.14	-0.13	1.00	0.28	7.85	4.77	1.54	-0.97	1.01	-2.23
Variable 2	10.52	5.60	4.41	4.62	12.31	9.55	5.76	2.51	2.54	1.59
Variable 3	18.27	10.82	11.17	11.80	18.25	17.14	11.84	8.39	10.00	7.61
Variable 4	16.24	10.98	10.46	11.22	18.07	16.77	11.58	8.62	10.56	8.74

In order to calculate the test statistic, each feature must receive a rank, assigned in ascending order ('bigger is better') according to the associated SN Saliency value. These ranks, along with the rank sum calculations (R_i) appear in Table 4.

Table 4. Feature Ranks for Design Point 9 - 0.1 - 0.1

Nodes	9	Learning Rate Step Size 0.1					Momentum Rate				0.1
	Replicates										
Feature	1	2	3	4	5	6	7	8	9	10	R_i
Noise	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	32.5	325
Noise 1	58	24	25	16	66	57	18	1	2	5	272
Noise 2	52	47	39	54	68	63	50	22	10	12	417
Noise 3	27	7	4	43	46	11	19	15	17	13	202
Noise 4	51	42	23	9	71	62	21	8	3	6	296
Bias	99	83	86	91	100	95	89	75	84	72	874
Variable 1	59	26	40	38	65	56	44	20	41	14	403
Variable 2	77	60	53	55	90	73	61	48	49	45	611
Variable 3	98	79	81	87	97	94	88	67	74	64	829
Variable 4	92	80	76	82	96	93	85	69	78	70	821

The ranks in Table 4 allow the calculation of the test statistic for design point 9 - 0.1 - 0.1. Similar calculations for every design point and for the overall design region are summarized in Table 5:

Table 5 Kruskal-Wallis H Test Statistic Values

α		Learning Rate Step Size								
0.05		0.1			0.5			0.9		
χ^2_α		Momentum Rate								
16.919		0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
Node	9	70.40	71.45	41.50	76.50	77.28	52.72	81.70	77.38	41.87
	18	88.15	85.98	17.76	80.82	67.68	49.98	82.54	78.43	39.94
	27	84.23	75.35	16.79	80.27	74.80	31.73	80.18	78.81	22.90
Overall					2696.293					

Please recall that due to the violation of the independence assumption, no inference may be drawn based upon these results. However, the large values of the test statistic at almost every design point supports the assertion that the distributions of the SN Saliency metric across all of the features are not identical.

The shaded value in Table 5 is the lowest value of the test statistic, and the only case where the test statistic does not exceed the critical value, while the underlined value in Table 5 corresponds to design point 9 - 0.1 - 0.1, which provides the values in Table 3 and Table 4.

Further evidence is provided by graphically comparing the rank distributions across all of the features. Figure 5 presents histograms of each "nonsalient" feature's rank across the entire design region while Figure 6 presents similar histograms for the "salient" features. A quick comparison of these histograms certainly suggests that the distributions of the ranks across the features are not identical, and since the ranks are based upon the SN Saliency values, the distributions of the SN Saliency values across the features are definitely not identical.

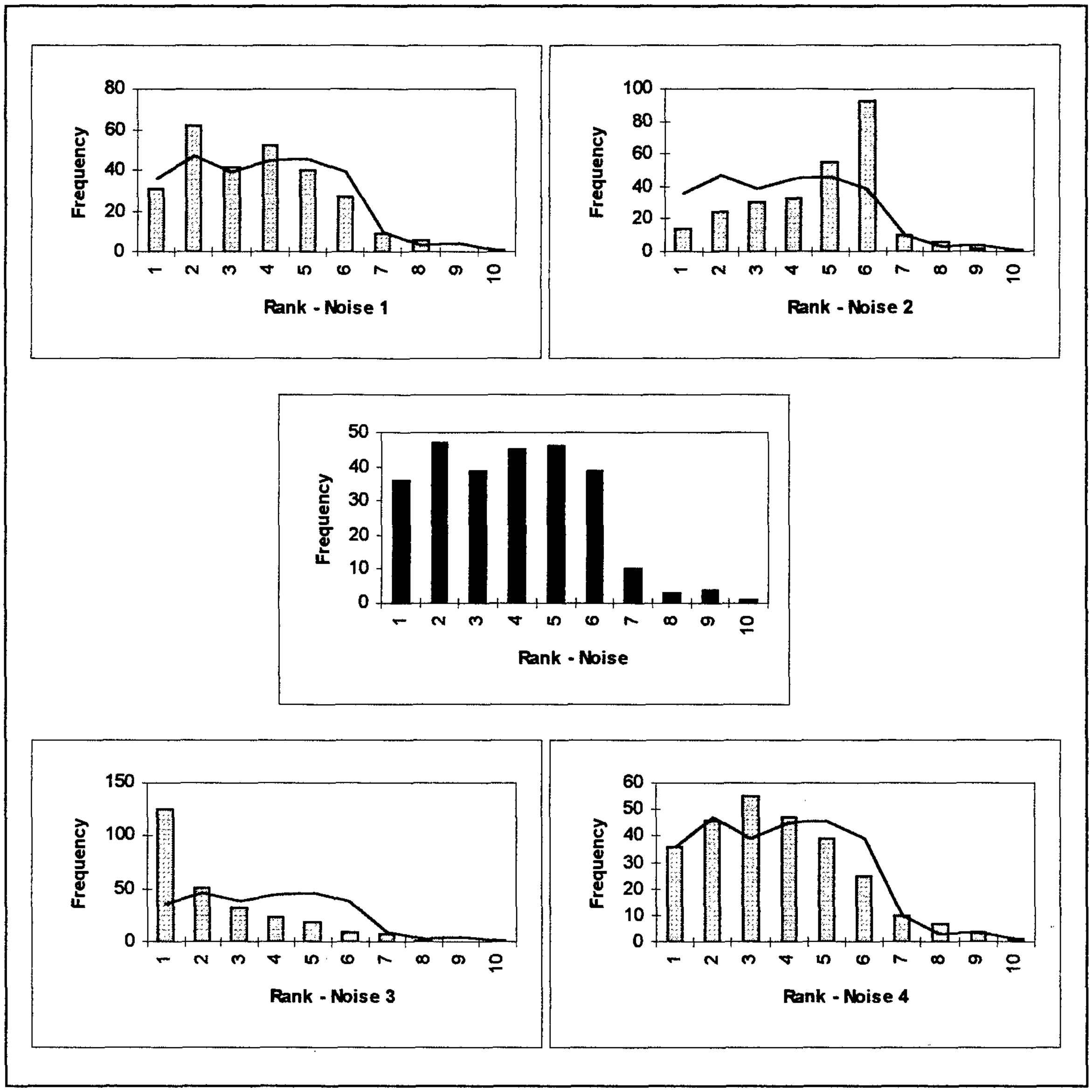

Figure 5. Rank Distributions of the "Nonsalient" Features

(2) If the distributions are different, how does each feature's distribution compare to that of the injected noise feature? Table 6 summarizes the Paired *t*-Test results. The rank distributions of the features shown in Table 6 are shifted to the right of the rank distribution for Noise, the injected noise feature. The rank distributions for the remaining

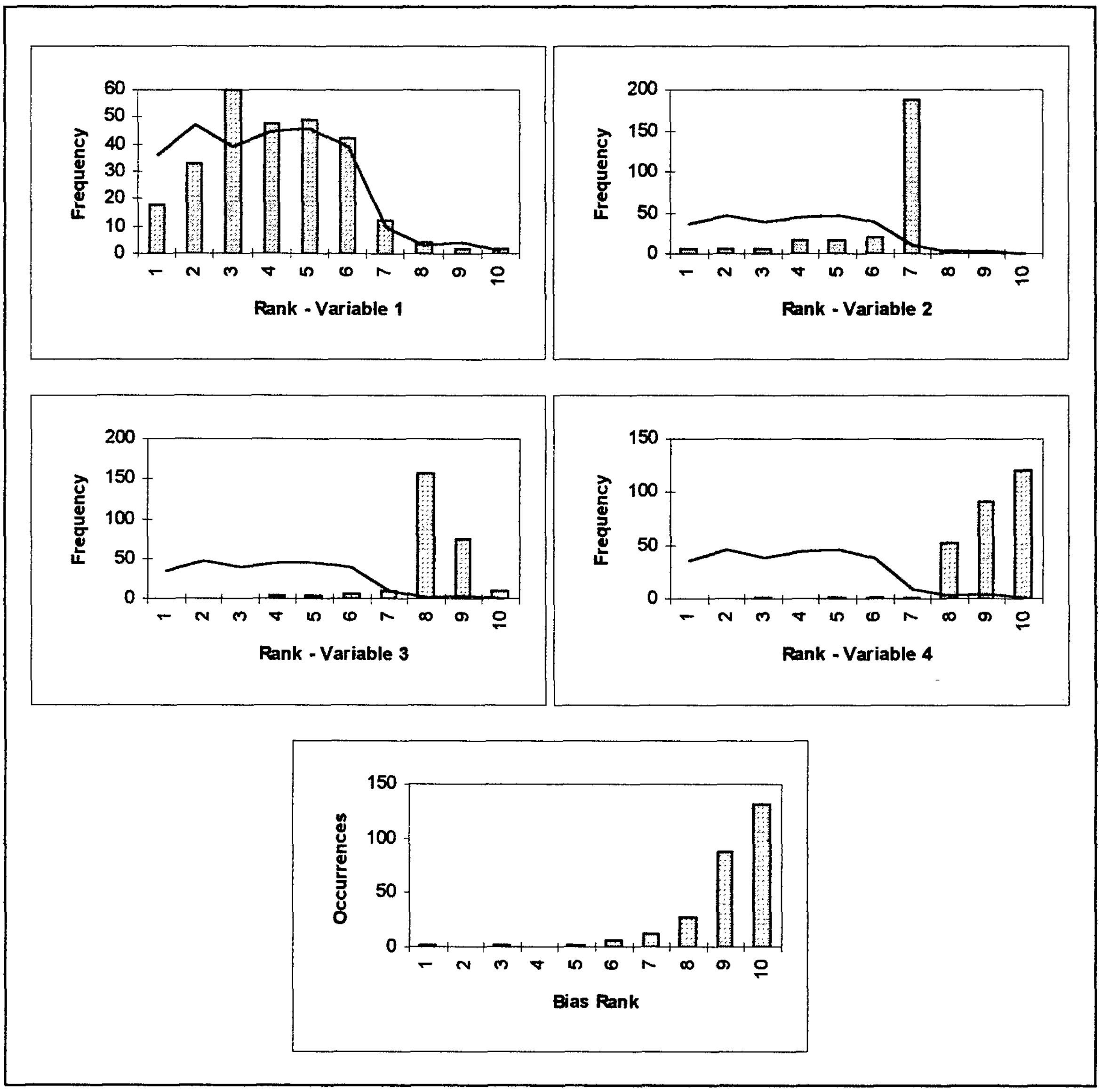

Figure 6 Rank Distributions of the "Salient" Features

Table 6. Paired *t*-Test Summary

Learning	Momentum	Number of Middle Nodes		
Rate	Rate	9	18	27
0.1	0.1	Variable 2 Variable 3 Variable 4	Variable 2 Variable 3 Variable 4 Noise 2	Variable 2 Variable 3 Variable 4
	0.5	Variable 2 Variable 3 Variable 4	Variable 2 Variable 3 Variable 4	Variable 2 Variable 3 Variable 4
	0.9	Variable 3 Variable 4	Variable 4	Variable 4
0.5	0.1	Variable 2 Variable 3 Variable 4	Variable 2 Variable 3 Variable 4 Noise 2	Variable 2 Variable 3 Variable 4
	0.5	Variable 2 Variable 3 Variable 4	Variable 2 Variable 3 Variable 4	Variable 2 Variable 3 Variable 4 Noise 2
	0.9	Variable 3 Variable 4	Variable 3 Variable 4 Noise 2	Variable 4
0.9	0.1	Variable 2 Variable 3 Variable 4 Noise 2	Variable 2 Variable 3 Variable 4 Noise 2	Variable 2 Variable 3 Variable 4
	0.5	Variable 2 Variable 3 Variable 4	Variable 2 Variable 3 Variable 4	Variable 2 Variable 3 Variable 4
	0.9	Variable 3 Variable 4	Variable 3 Variable 4	Variable 4

features are either collocated with or shifted to the left of the rank distribution for Noise.

Figure 7 provides further summary of the Paired *t*-Test results. This scree plot shows the number of design points, out of a total of 27, in which a given feature rejects

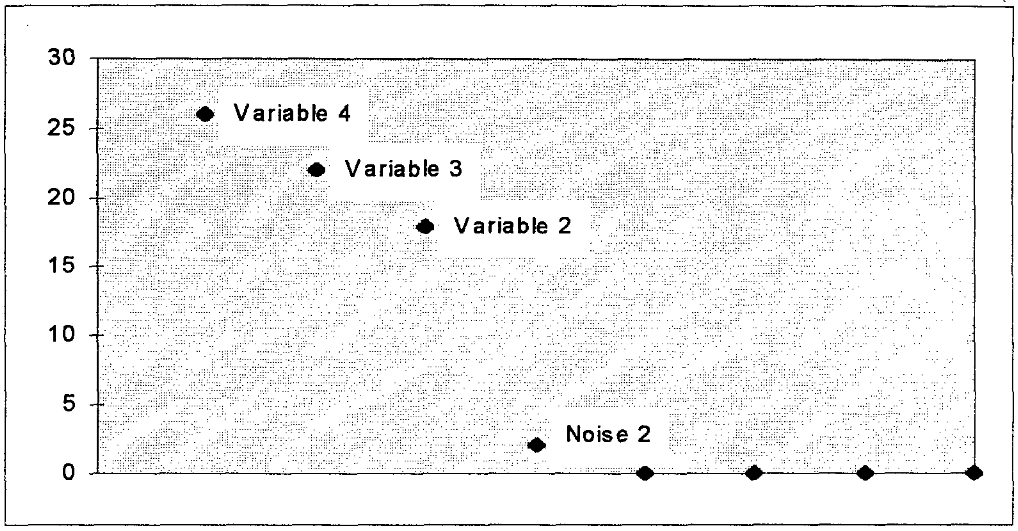

Figure 7. Scree Plot of Paired *t*-Test H_0 Rejections (by Feature)

the null hypothesis, resulting in a conclusion that the feature's rank distribution is shifted to the right of the injected noise feature's rank distribution. A high number of rejections suggests the feature is salient, while a low number of rejections suggests the feature is nonsalient. Only those features which generate any rejections are labeled.

(3) Are the feature ranks obtained from the SN saliency values consistent? Table 7 shows the results of Spearman's Rank Correlation Test for design point 9 - 0.1 - 0.1, while Table 8 summarizes the results across the entire design region. This summary shows the percentage of Spearman's Rank Correlation Tests, by design point, which reject H_0, leading to the conclusion that the paired ranks are consistent. These percentages should not be confused with a family confidence level, for each test is treated individually; none are treated as simultaneous tests. Nevertheless, high percentages do provide an indication that the ranks generated by the SN Saliency metric are fairly robust both within and

Table 7. Spearman's Rank Correlation Coefficients for Design Point 9 - 0.1 - 0.1

$\alpha = 0.05$	Middle Nodes			Learning Rate Step Size			Momentum Rate		
r_0	9			0.1			0.1		
0.564	Replicate								
Replicate	2	3	4	5	6	7	8	9	10
1	0.806	0.939	0.770	0.867	0.891	0.842	0.733	0.770	0.721
2		0.867	0.782	0.891	0.939	0.927	0.879	0.770	0.782
3			0.855	0.782	0.855	0.952	0.891	0.879	0.855
4				0.733	0.770	0.903	0.891	0.867	0.855
5					0.976	0.806	0.673	0.636	0.612
6						0.867	0.745	0.661	0.648
7							0.939	0.891	0.879
8								0.939	0.964
9									0.988

across the associated design points, while low percentages indicate that the associated design points might not produce robust ranks and ought to be avoided.

The high percentages associated with most design points in Table 8 suggest that, across and within these design points, the ranks are consistent. This increases the confidence that one run of the neural network in any of these design points will provide a reliable feature rank base upon the SN saliency metric.

Table 8. Percentage of Spearman's Rank Correlation Tests which reject H_0.

		Learning Rate Step Size								
		0.1			0.5			0.9		
		Momentum Rate								
		0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
Node	9	100	100	62.2	100	100	68.9	100	100	51.1
	18	100	100	15.6	100	100	68.9	100	97.8	53.3
	27	100	100	4.4	100	100	46.7	100	100	26.7
Overall					75.4					

(4) Does network architecture affect these results? A quick review of Tables 5, 6 and 8 reveals that the lowest H_0 rejection rates for each of these tests seem to be associated with a momentum rate of 0.9, suggesting that this momentum rate should be avoided when preparing for SN Saliency Screening.

Figures 8 provides graphical support of this suggestion. Each graph allows comparison of the rank distributions produced by a given level of each factor. For two of the factors, Number of Middle Nodes and Learning Rate Step Size, changing from one level to another changes the rank distribution only marginally. For Momentum Rate, however, changing to a level of 0.9 causes a significant shift in the rank distribution, resulting in a wider distribution. Wide distributions suggest that the neural network has difficulty determining the relative saliency of the associated feature, while narrow distributions suggest the neural network has little difficulty determining the saliency of the associated feature; therefore, narrow rank distributions are preferred, and the Momentum Rate plot suggests that a momentum rate of 0.9 should be avoided, which is consistent with the results generated by Kruskal-Wallis' H-Test, the Paired *t*-Test, and Spearman's Rank Correlation Test. Although Figure 4.4 only shows Noise, the results are similar for the other features.

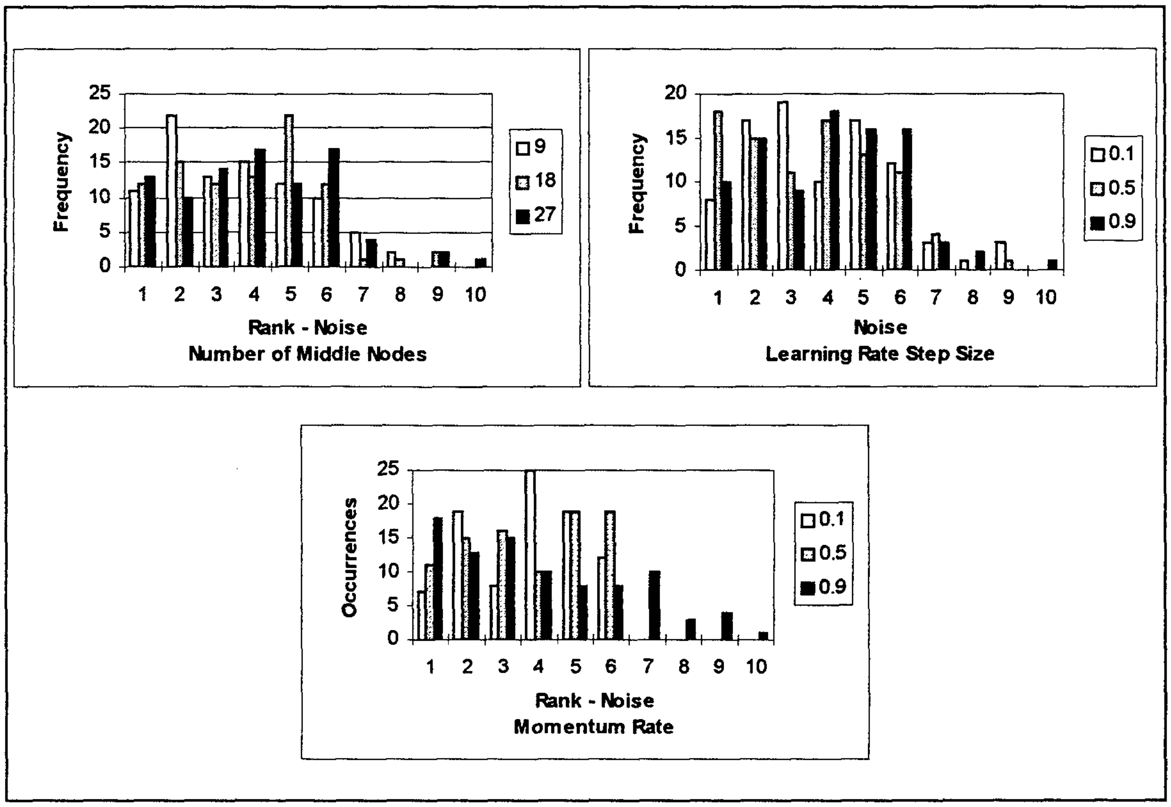

Figure 8. Design Factor Effect on Rank Distribution

XOR Problem.

(1) Are the distributions of the SN Saliency values identical for all features? The calculation of the Kruskal-Wallis H Test Statistic begins with the SN Saliency values. Table 9 shows these values at design point 6 - 0.1 - 0.1. Note that Noise 1, Noise 2 and Noise 3 tend to have the lower values, indicating low relative saliency, while Variable 1 and Variable 2 tend to have the higher values, indicating high relative saliency.

Table 9 SN Saliency Values for Design Point 6 - 0.1 - 0.1

Nodes	6	Learning Rate Step Size 0.1					Momentum Rate 0.1			
	Replicates									
	1	2	3	4	5	6	7	8	9	10
Noise	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Noise 1	-5.33	3.60	2.42	-4.71	-4.07	3.16	-5.15	-6.04	-4.35	-2.81
Noise 2	-6.44	4.00	-5.80	-1.04	-7.55	6.84	-0.20	1.65	-3.83	6.01
Noise 3	-3.54	-1.70	1.49	-0.46	-3.55	3.40	1.93	-8.82	-6.80	-6.12
Bias	8.32	15.53	13.76	11.36	7.82	12.16	11.63	5.83	6.74	9.74
Variable 1	9.07	15.48	14.52	12.44	8.65	12.73	12.50	7.08	5.98	10.27
Variable 2	8.49	15.78	13.62	11.66	7.71	13.21	12.90	7.65	6.42	11.01

The next step in the test statistic calculation is to convert the SN Saliency values to ranks, and then to sum these ranks for each feature. Table 10 contains these ranks and R_i, the rank sum for the *i*-th feature. Note that Noise, the injected noise feature, receives an average rank for each replicate in order to break the ties caused by its constant SN Saliency value. Remember that the features are ranked in ascending order ('bigger is better') and across all ten replicates in the design point. Note how much larger the ranks

Table 10. Feature Ranks for Design Point 6 - 0.1 - 0.1

Nodes	6	Learning Rate Step Size				0.1	Momentum Rate				0.1
	Replicates										
	1	2	3	4	5	6	7	8	9	10	R_i
Noise	25.5	25.5	25.5	25.5	25.5	25.5	25.5	25.5	25.5	25.5	255
Noise 1	8	37	34	10	12	35	9	6	11	16	178
Noise 2	4	38	7	18	2	44	20	32	13	41	219
Noise 3	15	17	31	19	14	36	33	1	3	5	174
Bias	49	69	66	56	48	59	57	39	43	53	539
Variable 1	52	68	67	60	51	62	61	45	40	54	560
Variable 2	50	70	65	58	47	64	63	46	42	55	560

and rank sums are for Variable 1 and Variable 2 compared to Noise, Noise 1, Noise 2 and Noise 3.

Table 11 shows the test statistic values for each design point in the entire design region. The underlined value belongs to design point 6 - 0.1 - 0.1 used in Table 9 and

Table 11. Kruskal-Wallis H-Test Statistic Values

α		Learning Rate Step Size								
0.05		0.1			0.5			0.9		
χ^2_α		Momentum Rate								
12.592		0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
Node	6	50.82	49.13	15.30	50.05	49.75	43.28	50.57	48.66	40.14
	12	57.06	52.94	22.27	53.35	50.13	48.83	53.71	52.64	37.00
	18	49.29	59.67	32.77	53.15	55.32	37.56	53.12	55.88	26.57
Overall					1883.49					

Table 10. The shadowed value shows the lowest value of the test statistic across the entire design region. This lowest value (15.30) is still greater than the critical value (12.592), and even though no statistical inference may be drawn from this test due to the violation of the independence assumption, the magnitude of the test statistic values throughout the design region indicates that the rank distributions are not identical across the features.

Figure 9 and Figure 10 provide further evidence in support of this assertion. A quick comparison of the distributions in these two figures reveals a significant difference

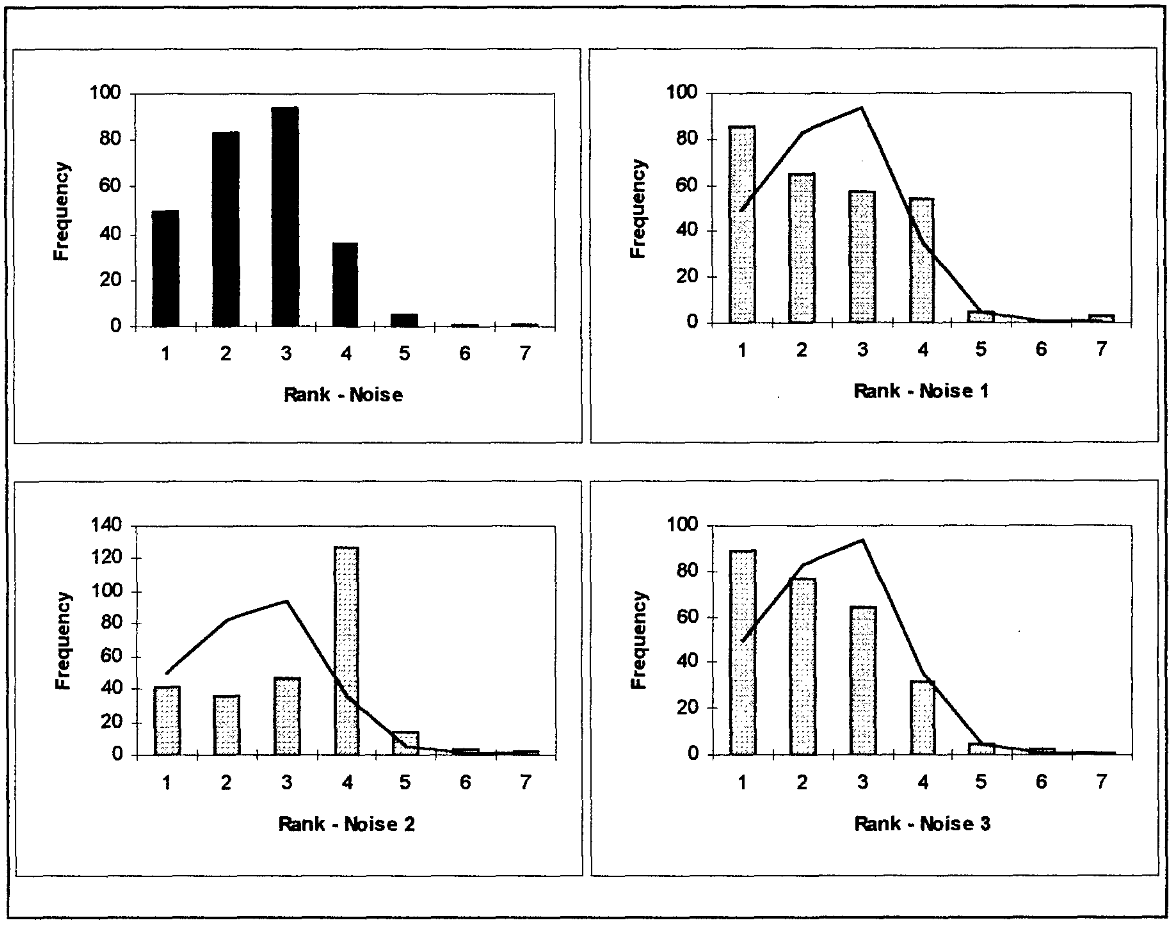

Figure 9. Rank Distributions of "Nonsalient" Features

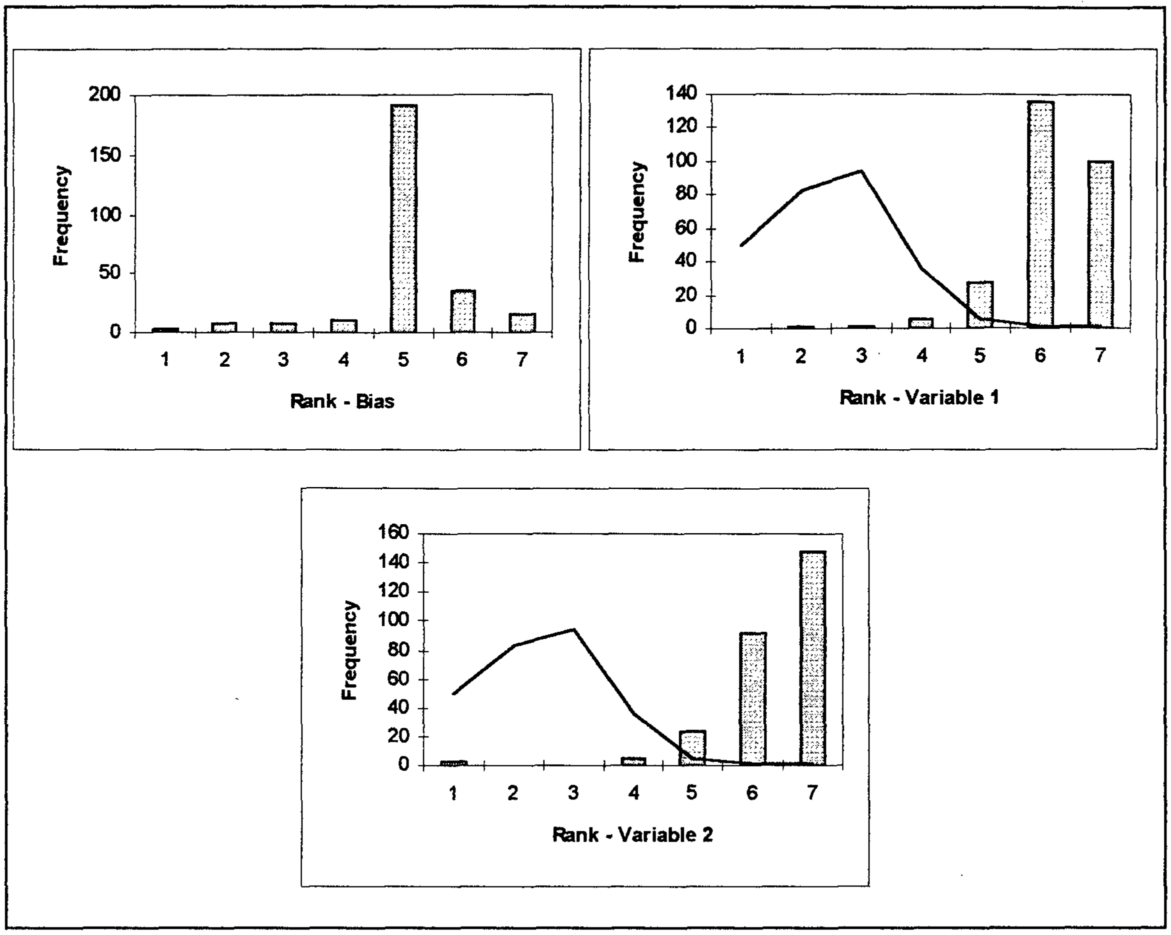

Figure 10. Rank Distributions of "Salient" Features

among the locations of the rank distributions, especially between the "nonsalient" and the "salient" features.

(2) If the distributions are different, how does each feature's distribution compare to that of the injected noise feature? Table 12 provides a summary of the Paired *t*-Tests run in each design region. Rejecting the null hypothesis associated with these upper-tailed *t*-tests supports the conclusion that the associated feature's rank distribution is shifted to the right of the injected noise feature's rank distribution. This shift to the right indicates

Table 12. Paired *t*-Test Summary

Learning Rate Step Size	Momentum Rate	Number of Middle Nodes		
		6	12	18
0.1	0.1	Variable 1 Variable 2	Variable 1 Variable 2	Variable 1 Variable 2
	0.5	Variable 1 Variable 2	Variable 1 Variable 2	Variable 1 Variable 2
	0.9		Variable 2	Variable 1 Variable 2
0.5	0.1	Variable 1 Variable 2	Variable 1 Variable 2	Variable 1 Variable 2
	0.5	Variable 1 Variable 2	Variable 1 Variable 2	Variable 1 Variable 2
	0.9	Variable 1 Variable 2	Variable 1 Variable 2 Noise 2	Variable 1 Variable 2
0.9	0.1	Variable 1 Variable 2	Variable 1 Variable 2	Variable 1 Variable 2
	0.5	Variable 1 Variable 2	Variable 1 Variable 2	Variable 1 Variable 2
	0.9	Variable 1 Variable 2	Variable 1 Variable 2	Variable 1 Variable 2

that the associated feature is more salient than Noise, the injected noise feature. Table 12 shows the features in each design point which rejected the null hypothesis.

The rank distributions for Variable 1 and Variable 2 are consistently shifted to the right of the rank distribution for Noise, suggesting that Variable 1 and Variable 2 are salient features. The consistent failure of the other features to reject the null hypothesis suggests that these remaining features are not salient (note: the Bias feature is not included in these tests).

Additional evidence indicating differences in rank distribution locations is provided by a quick review of Figures 9 and 10. The graphs for Noise 1, Noise 2, Noise 3, Variable 1 and Variable 2 each contain a secondary linear plot which corresponds to the rank distribution of Noise. A quick comparison of each feature's primary histogram plot to the secondary linear plot shows that Variable 1 and Variable 2 have definitely shifted to the right, Noise 1 and Noise 3 have not shifted to the right, and Noise 2 is inconclusive.

Figure 11 provides a scree plot of the number of null hypothesis rejections attained out of 27 total tests by each feature. This plot divides the features into two distinct groups. The first group, containing Variable 1 and Variable 2, is characterized by a high number of rejections, strongly suggesting saliency. The second group contains Noise 1, Noise 2 and Noise 3 (Noise 1 and Noise 3 are not labeled since they produced no rejections), each of which produced few or no rejections, strongly suggesting nonsaliency.

(3) Are the feature ranks obtained from the SN Saliency values consistent? Spearman's Rank Correlation Test investigates this issue of consistency. Table 13 shows

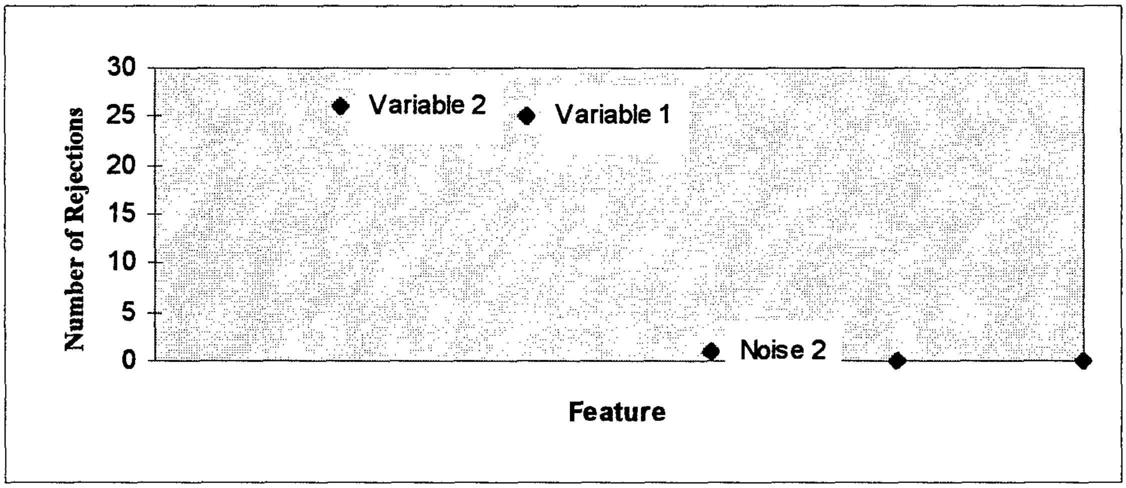

Figure 11. Scree Plot of Paired *t*-Test H_0 Rejections (by Feature)

Table 13. Spearman's Rank Correlation Coefficients for Design Point 6 - 0.1 - 0.1

$\alpha = 0.05$	Middle Nodes			Learning Rate Step Size			Momentum Rate		
r_0	6			0.1			0.1		
0.714	Replicate								
Replicate	2	3	4	5	6	7	8	9	10
1	0.571	0.821	0.964	0.964	0.643	0.893	0.714	0.714	0.714
2		0.607	0.607	0.536	0.857	0.643	0.929	0.857	0.929
3			0.714	0.857	0.643	0.679	0.571	0.607	0.571
4				0.929	0.714	0.929	0.786	0.75	0.786
5					0.571	0.821	0.643	0.75	0.643
6						0.821	0.857	0.643	0.857
7							0.75	0.679	0.75
8								0.857	1
9									0.857

the correlation coefficients calculated for each pair of replicates in design point 6 - 0.1 - 0.1. The shaded values do not exceed the critical value (0.714), leading to the conclusion that the paired replicates involved in the particular test do not provide consistent feature ranks. The critical value shown above corresponds to a single test. No family confidence level is offered for making simultaneous inferences from more than one test. On the other hand, the percentage of these tests which reject H_0, both within a given design point and across design points, provides some indication of the consistency of these ranks. Higher rejection percentages support the conclusion that the feature ranks are consistent, while lower rejection percentages suggest that the ranks are not consistent.

With this in mind, Table 14 shows these rejection percentages both within and across design points. Overall, the percentages seem to be lower than those produced by the Fisher Iris Classification problem. The scree plot in Figure 11 may provide a reason.

Table 14 Percentage of Spearman's Rank Correlation Tests which Reject H_0

		Learning Rate Step Size								
		0.1			0.5			0.9		
		Momentum Rate								
		0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
Node	6	57	100	23	90	62	52	71	90	43
	12	95	100	14	81	71	67	81	76	33
	18	71	100	29	76	81	43	86	76	24
Overall					64.296					

Though the distinction between the two groups of features (based on the number of Paired *t*-Test rejections of H_0) seems fairly significant, the distinction between the features within the groups may be insignificant, perhaps due to similar *a priori* saliencies within each group. If the neural network does not distinguish well between the saliencies of Noise 1, Noise 2 and Noise 3, it may well produce inconsistent ranks of these three features, leading to a low percentage of H_0 rejections by Spearman's Rank Correlation Test.

Most likely, the SN saliency metric ranks Variable 1 and Variable 2 consistently higher than Noise 1, Noise 2 and Noise 3, which is the desired consistency. Table 15 shows the percentage of replications within each design point and overall in which Variable 1 and Variable 2 are ranked higher than Noise, Noise 1, Noise 2 and Noise 3. The high percentage rates associated with most of the design points suggest that the SN

Table 15 Percentage of Replications which rank Variable 1 and Variable 2 above Noise, Noise 1, Noise 2 and Noise 3

		Learning Rate Step Size								
		0.1			0.5			0.9		
		Momentum Rate								
		0.1	0.5	0.9	0.1	0.5	0.9	0.1	0.5	0.9
Node	6	100	100	50	100	100	90	100	100	90
	12	100	100	50	100	100	90	100	100	100
	18	100	100	60	100	100	100	100	100	90
Overall					93.3					

saliency metric consistently ranked the 'salient' features higher than the 'nonsalient' features, which is the desired result.

(4) Does network architecture affect results? Figure 12 shows the changes in the rank distribution of Noise due to changes in the level of one of the three neural network design factors. Each graph corresponds to a different design factor (e.g. Number of Middle Nodes), and the three plots within the graph are each associated with a different level of that design factor (e.g. 6 nodes, 12 nodes, 18 nodes). Significant differences caused by changing the level of the design factor are easily recognized by comparing the plots associated with the three levels of that design factor.

Of the three design factors (Number of Middle Nodes, Learning Rate Step Size and Momentum Rate), setting the momentum rate to 0.9 appears to cause the biggest shift in the rank distribution, although this shift may or may not be problematic. The tightest distributions seem associated with neural networks which use 18 middle nodes, a learning rate step size of 0.5 and avoid a momentum rate of 0.9.

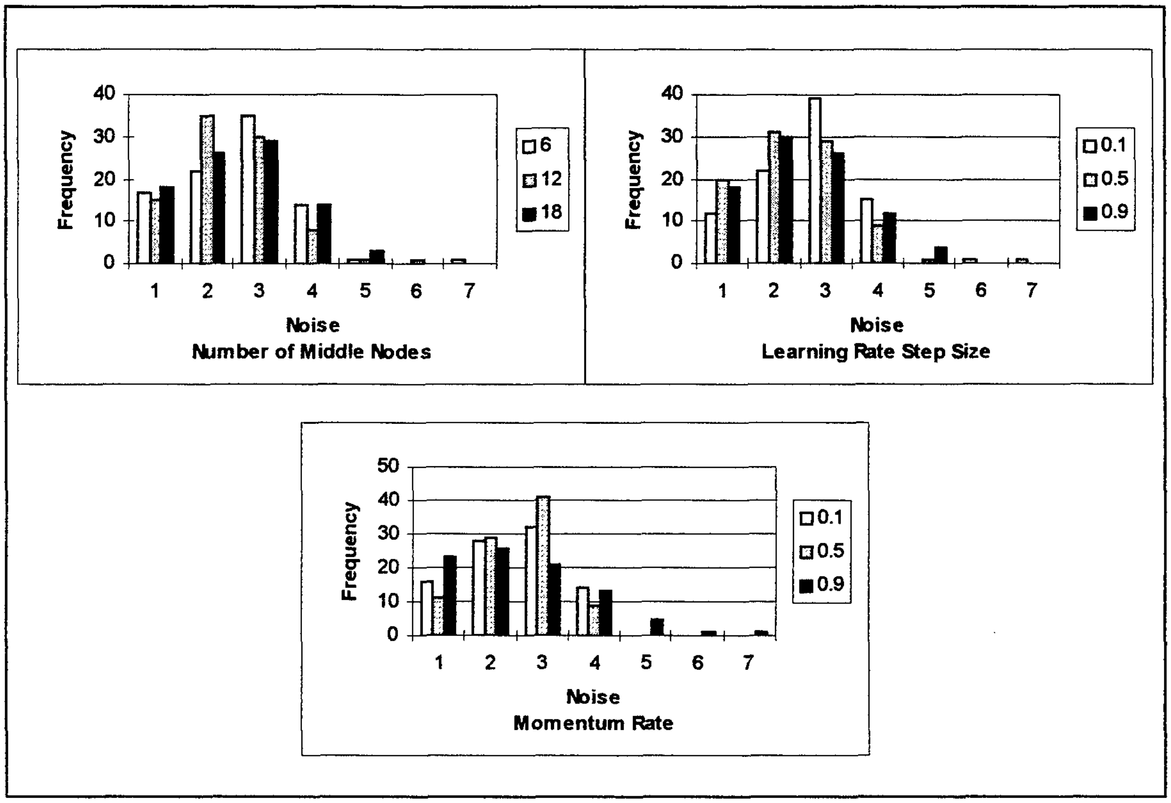

Figure 12. Design Factor Effect on Rank Distribution

The three statistical tests each provide some support for avoiding a high momentum rate. A quick review of Tables 11, 12, 14 and 15 reveals that the lowest test statistic values consistently correspond to the highest momentum rate, while in Table 12, the only spurious results of the Paired *t*-Test correspond to the highest momentum rate. All of this seems to suggest that the neural network has more difficulty sorting features according to saliency at the highest momentum rate, and that this momentum rate should be avoided.

SN Saliency Screening

This section explores the robustness of the SN Saliency Screening method, both within and across design points, by examining the results obtained when applying the method to two example problems: the Fisher Iris Classification problem and the XOR problem. This section is divided into two parts, each devoted to a particular example problem.

The results obtained through application of the SN saliency screening method to each of the two example problems indicate that the method is robust both within and across design points. Each SN saliency screening run identified similar (in the Fisher Iris Classification problem) or identical (in the XOR problem) subsets of salient features. The variation in the subset of salient features identified for the Fisher Iris Classification problem indicates redundancy among the salient features, which allows the neural network to retain either of the two most highly salient features with negligible effect on the classification error rate.

Fisher Iris Classification Problem. In order to explore the robustness of the SN saliency screening method across design points, one screening run, using the method described in Chapter 3, is accomplished at three separate design points: 9 - 0.1 - 0.1, 18 - 0.5 - 0.5 and 27 - 0.1 - 0.9. Figures 13, 14 and 15 summarize the saliency screening by showing when each feature is removed and the classification error rate's response to that removal.

The classification error chart shows the test set error rate, for it provides a more accurate approximation of the actual error rate. A lower classification rate is better than a larger classification error rate. Notice how the classification error rate is reasonable stable until approximately 1600 epochs, at which time it leaps up significantly. A feature is removed about every 200 epochs, and this point corresponds to the removal of the eighth, i.e. the last, of the original features in Fisher's Iris data set. This suggests that the neural network only needs one of the features to maintain a reasonably small classification error rate. The graph directly beneath the classification error rate graph shows which of the features is retained until last.

The lower graph shows the signal-to-noise ratios, or SN saliency values, for the eight original features, Noise and Bias. The legend shows the eight original features in the order in which they are removed. For example, Figure 13 shows that Noise 2 is removed first, Noise 3 is removed second, and so on. The feature which is removed at a given point has the lowest signal-to-noise ratio of all of the candidate features at that point. Remember, bigger is better for signal-to-noise ratios. Bias and Noise are not removed during the saliency screening. Noise is retained for calculation of the SN saliency value, while Bias is supplied directly by the neural network and is not subject to removal.

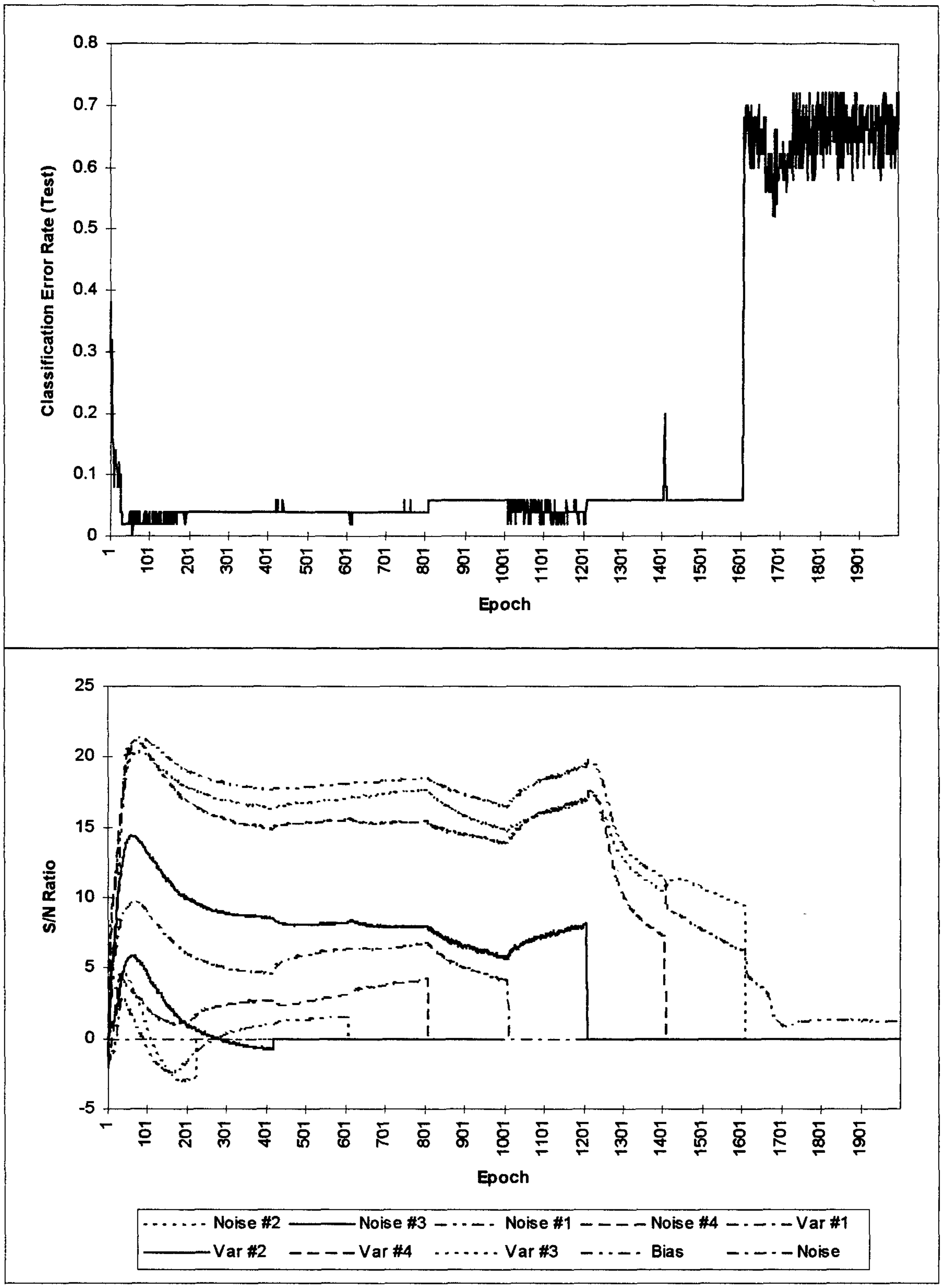

Figure 13. Classification Error Rate and Signal-to-Noise Ratios
(Design Point 9 - 0.1 - 0.1)

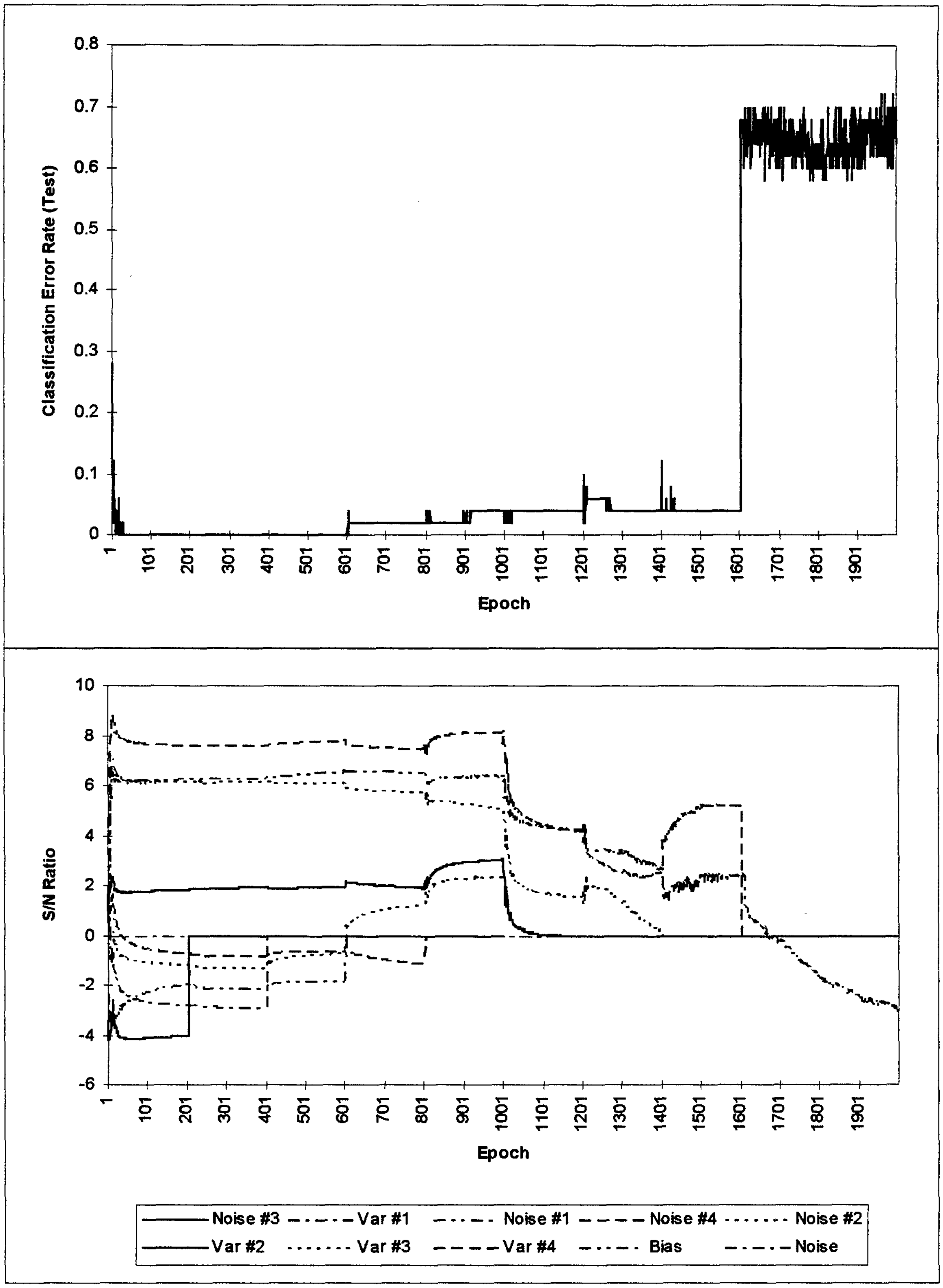

Figure 14. Classification Error Rate and Signal-to-Noise Ratios
(Design Point 18 - 0.5 - 0.5)

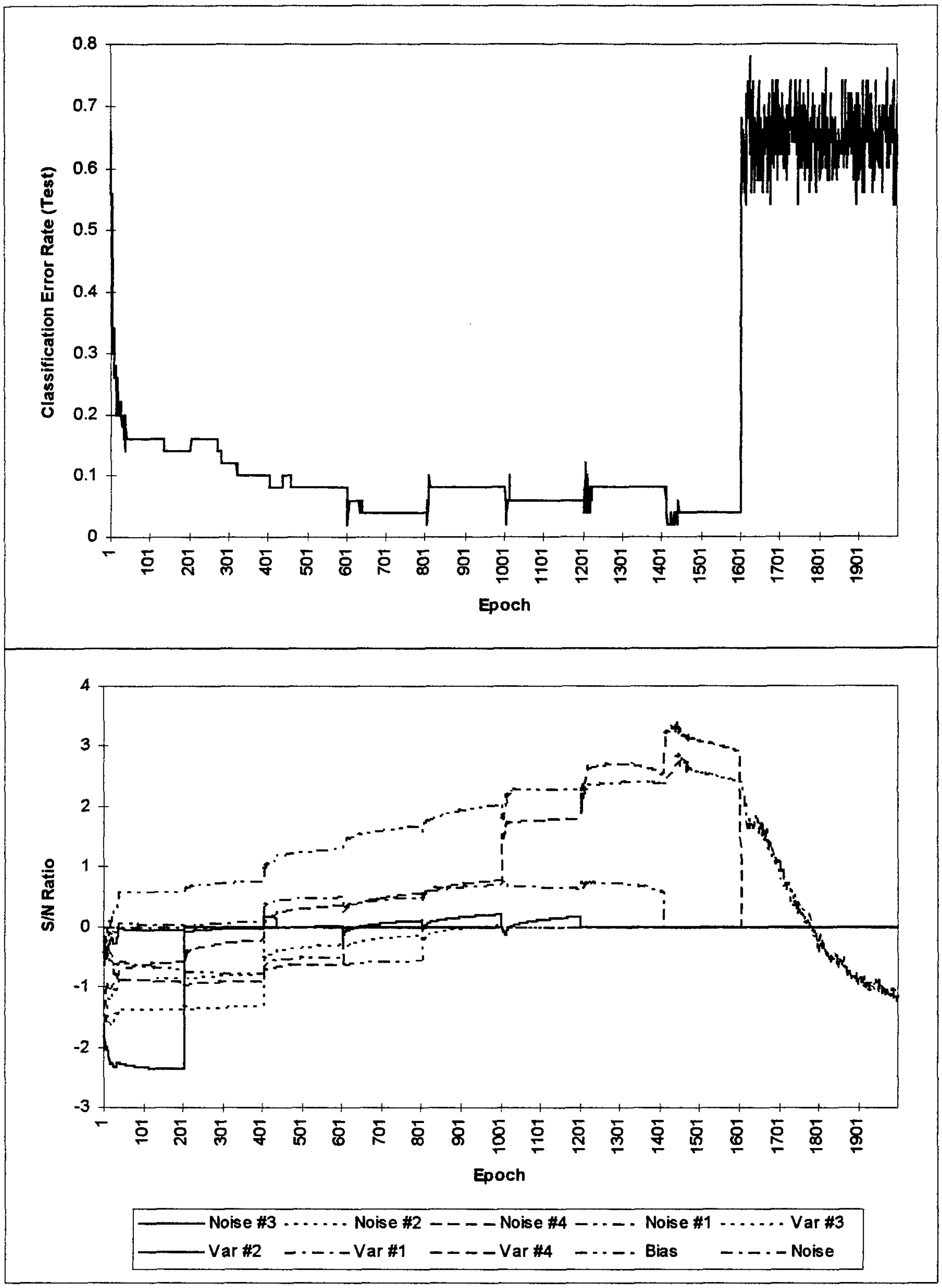

Figure 15. Classification Error Rate and Signal-to-Noise Ratios
(Design Point 27 - 0.1 - 0.9)

In comparing these three saliency screenings, note that the classification error rate reacts fairly consistently to feature removal across the three design points; the error rate is not significantly altered until the last of the original eight features is removed. Notice the difference in the magnitudes of the signal-to-noise ratios between design points. This complicates the problem of trying to identify salient features based simply upon the magnitude of the signal-to-noise ratios. However, even if the magnitudes of the signal-to-noise ratios vary significantly from design point to design point, the feature ranks provided by these values seem fairly consistent across the three design points. This supports a saliency screening method based upon the relative rank provided by the saliency metric.

Design point 9 - 0.1 - 0.1 recommends retaining Variable 3, while the other two points recommend retaining Variable 4. Why the difference, and why do they not recommend keeping both features, since previous tests seem to indicate that both features, as well as Variable 2 and possible Noise 2, are salient? Table 16 shows that Variable 3 and Variable 4 are highly correlated with each other and have similar linear correlations with the classification groups. Basically, the neural network receives the same information, the same value, from each feature; in other words, they are redundant. Hence, the neural network needs to retain only one of these two features to achieve similar classification accuracy.

Variable 2, on the other hand, is only moderately correlated with the second classification group. The high correlation of Variable 3 and Variable 4 with the first and third classification groups may be more valuable for accurately classifying an iris into the second group than the information given by Variable 2. Variable 3 and Variable 4 accurately predict when an iris does not belong to the first and third groups, in which case, by default, it must belong to the remaining group. If the classifier created by the neural network is able to deduce this , then the information provided by Variable 3 and Variable 4 is definitely more valuable than that provided by Variable 2 for classifying irises into the second group. The suggestion that a neural network operates this way qualifies as pure conjecture, although the suggestion does provide a convenient explanation.

Table 16. Correlation of Features and Classification Groups

	Var 1	Var 2	Var 3	Var 4	Noise 1	Noise 2	Noise 3	Noise 4	Group 1	Group 2	Group 3
Noise	-0.09	0.03	-0.11	-0.12	-0.01	0.07	-0.12	0.05	-0.14	0.07	0.07
Var 1		-0.12	0.87	0.82	0.06	0.00	0.11	-0.09	0.64	0.08	-0.72
Var 2			-0.43	-0.37	0.06	-0.10	-0.03	0.07	-0.14	-0.47	0.60
Var 3				0.96	0.04	0.03	0.08	-0.12	0.72	0.20	-0.92
Var 4					0.07	0.00	0.08	-0.10	0.77	0.12	-0.89
Noise 1						0.06	0.01	-0.06	-0.01	0.06	-0.05
Noise 2							-0.04	-0.05	-0.04	0.12	-0.08
Noise 3								-0.06	0.11	-0.03	-0.08
Noise 4									-0.14	0.06	0.08

In order to examine the robustness of the SN Saliency Screening method within a design point, ten saliency screenings are accomplished at design point 9 - 0.5 - 0.5. Figure 16 shows the average signal-to-noise ratios generated by these runs. The legend shows the average order in which the features are removed. This order is consistent with those produced by the single screenings at design points 9 - 0.1 - 0.1, 18 - 0.5 - 0.5 and 27 - 0.1 - 0.9. A feature is removed approximately every 200 epochs, and when a feature is removed, its SN saliency value goes to zero.

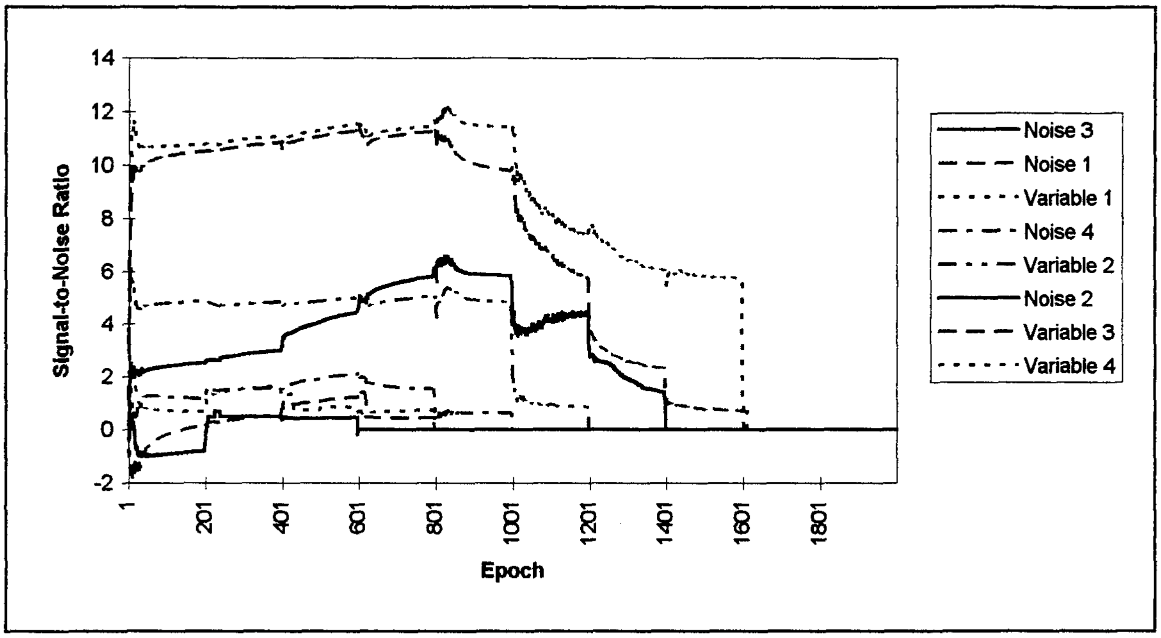

Figure 16. Average Signal-to-Noise Ratios
(Design Point 9 - 0.5 - 0.5, 10 Replicates)

Figure 17 shows the average classification error rate generated during these ten runs. This average is calculated for the epoch just prior to feature removal. Since eight features are removed, eight averages are calculated. These eight averages are connected by lines for readability and trend analysis. The lines between the eight points do not represent averages. The dashes immediately above and below the eight points represent adding and subtracting one standard deviation to the average. This range does not represent a confidence interval. However, the relative width of this range indicates relative variance of the classification error rate at each point of feature removal.

This plot is similar to the classification error rate plots for the previous three design points. Feature removal generates no significant change in the classification error

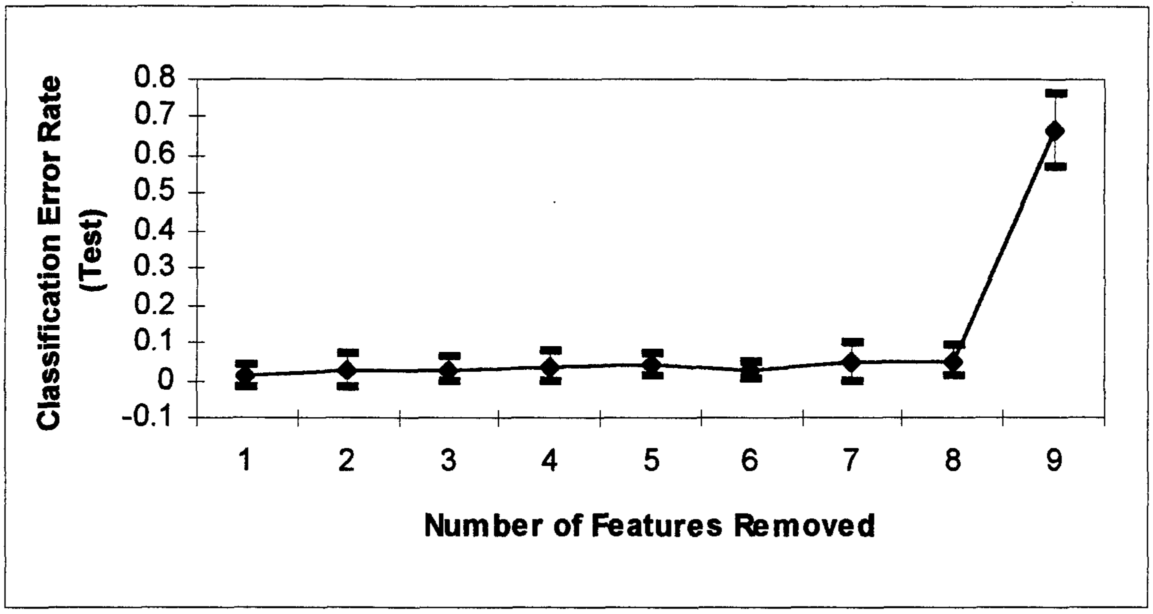

Figure 17. Average Classification Error Rate at Feature Removal
(Design Point 9 - 0.5 - 0.5, 10 Replicates)

rate until the last of the eight original features is removed. (Note: no ninth feature is removed; this point corresponds to the end of training.)

Figure 18 shows the average feature rank during these ten replications. The rank determines when the feature is removed. Once again, bigger is better, and the rank's range starts at one (first feature removed) and ends at eight (last feature removed). As expected, Variable 4 and Variable 3 occupy the two highest spots, with Variable 4 mildly preferred over Variable 3. Once again, the dashes represent adding and subtracting one standard deviation to the average rank. The relative width of the resultant interval provides some indication as to the certainty with which the neural network assesses the feature's relative saliency. The smallest interval belongs to Variable 4, while the largest belongs to Noise 4, which indicates that the neural network is much more certain about Variable 4's rank, and hence its saliency, than it is about Noise 4's rank.

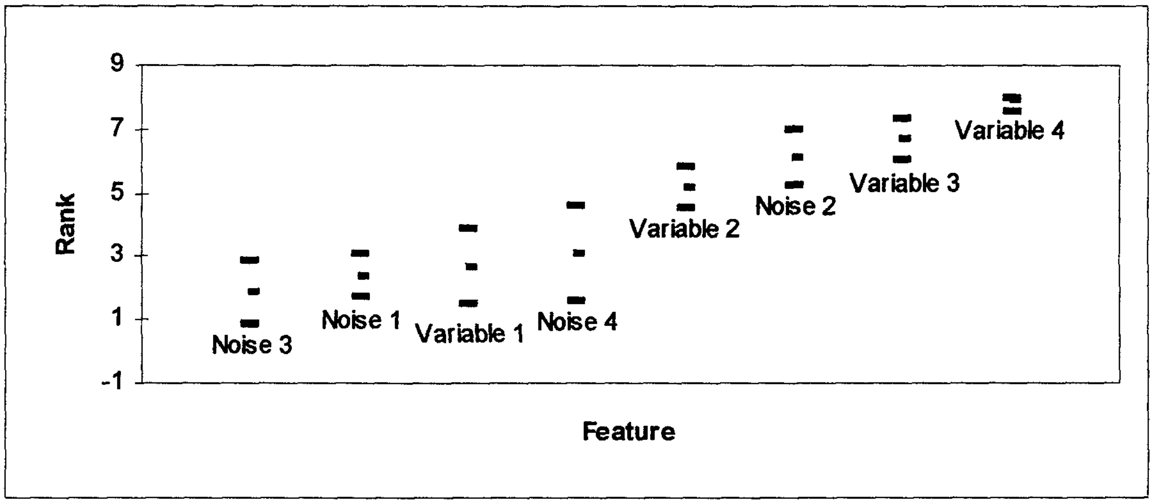

Figure 18. Average Feature Rank (Design Point 9 - 0.5 - 0.5, 10 Replications)

The XOR Problem. In order to investigate the robustness of the SN saliency screening method within and across design points, three single screenings are accomplished at design points 6 - 0.1 - 0.1, 12 - 0.5 - 0.5 and 18 - 0.9 - 0.9. Figures 19, 20 and 21 plot the classification error rate and average signal-to-noise ratios produced by each screening.

A quick scan of these three graphs reveals that the removal of either Variable 1 or Variable 2 causes a significant increase in the classification error rate, indicating that both of these features should be retained as salient. The SN saliency values tend to divide the features into two groups, one containing Variable 1 and Variable 2, and the second

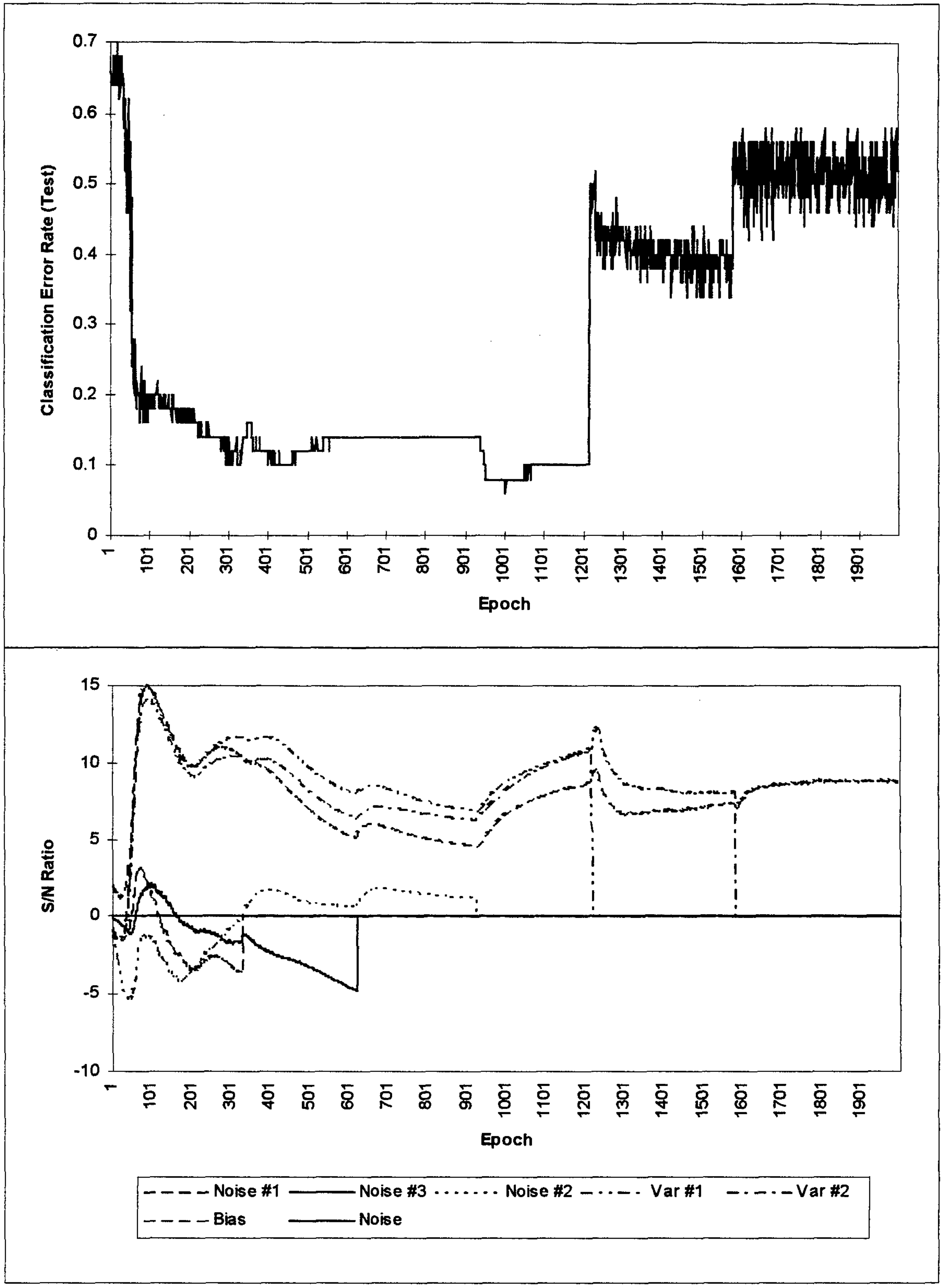

Figure 19. Classification Error Rate and Signal-to-Noise Ratios
(Design Point 6 - 0.1 - 0.1)

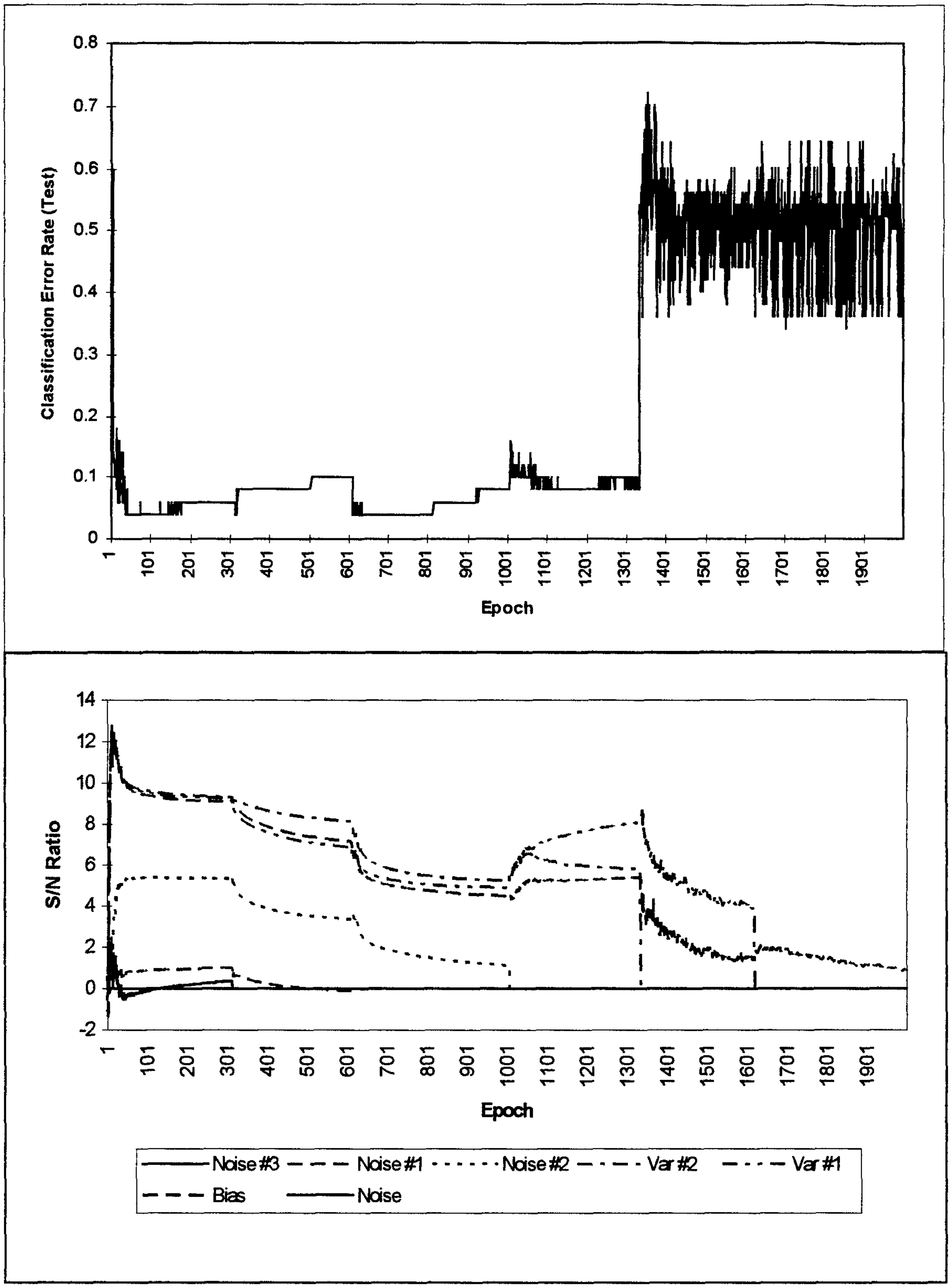

Figure 20. Classification Error Rate and Signal-to-Noise Ratios
(Design Point 12 - 0.5 - 0.5)

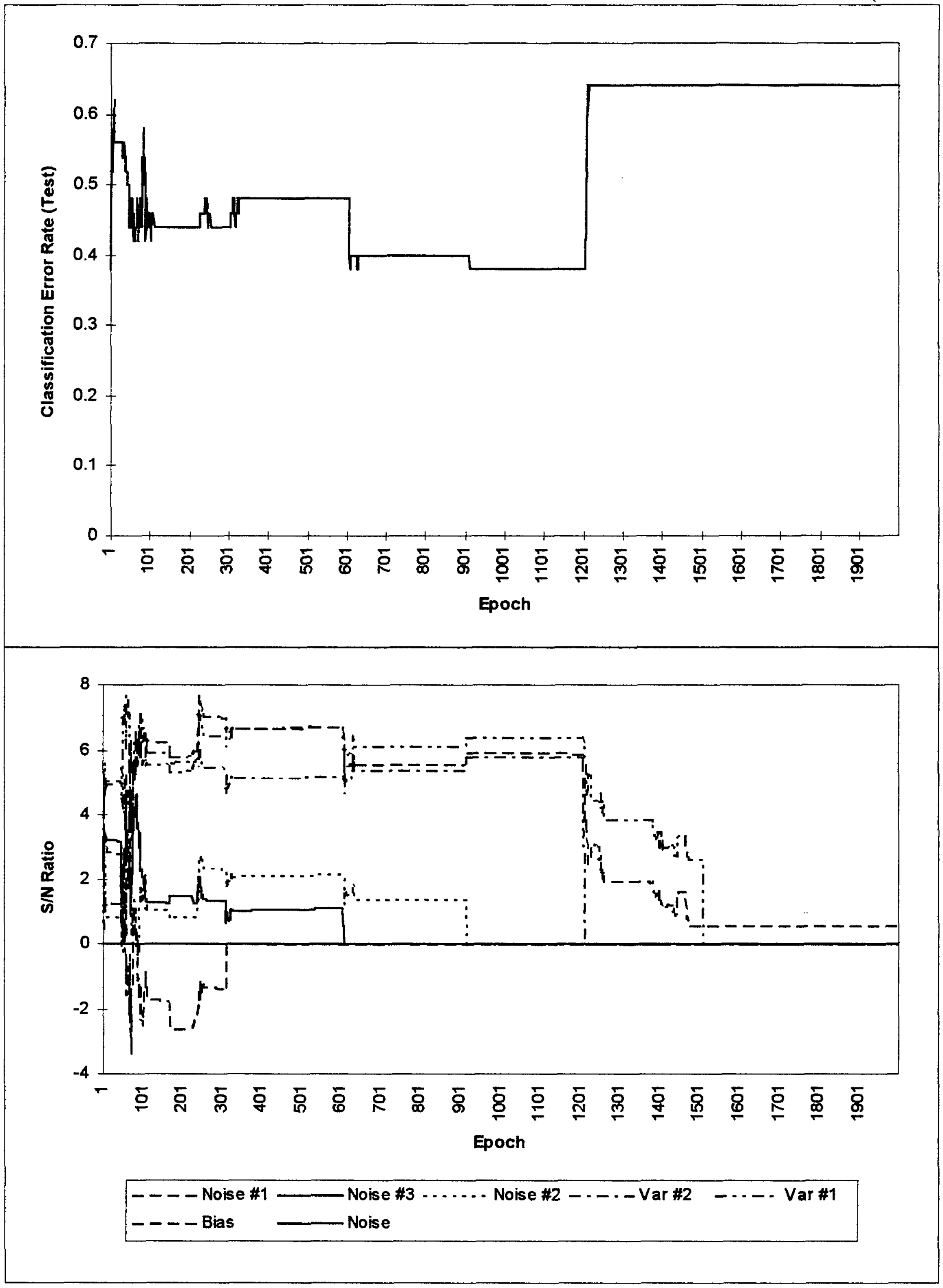

Figure 21. Classification Error Rate and Signal-to-Noise Ratios
(Design Point 18 - 0.9 - 0.9)

containing Noise 1, Noise 2 and Noise 3. Since the removal of any of the features in the second group caused no significant increase in the classification error rate, they are identified as nonsalient.

Figure 22 shows the results produced by ten saliency screenings at the design point 12 - 0.3 - 0.7. The average signal-to-noise ratios shown here are consistent with those shown in Figures 19, 20 and 21.

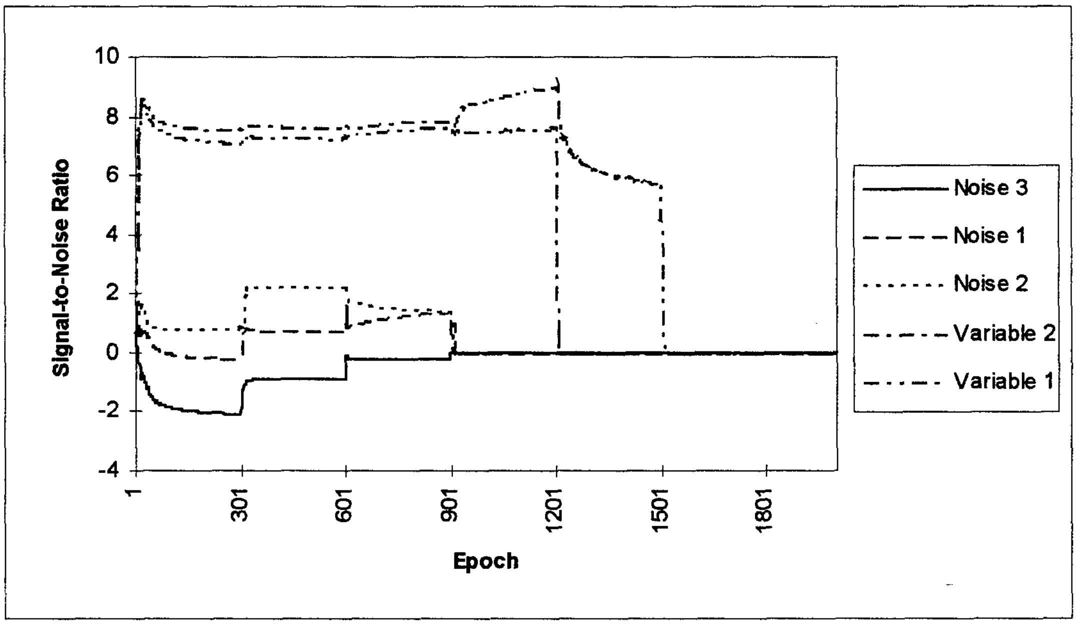

Figure 22. Average Signal-to-Noise Ratios
(Design Point 12 - 0.3 - 0.7, 10 Replications)

Figure 23 shows the average classification error rates at feature removal for the ten screening runs. This plot shows how well-behaved the XOR problem is. Removal of the nonsalient features causes a decreasing trend in the average classification error rate; i.e. the neural network is becoming more accurate. Furthermore, the variability of the classification error rate, as indicated by the interval bounded by the dashes, decreases throughout the removal of the nonsalient features. When the first of the two salient features is removed, the classification error rate increases significantly, both in magnitude and in variability.

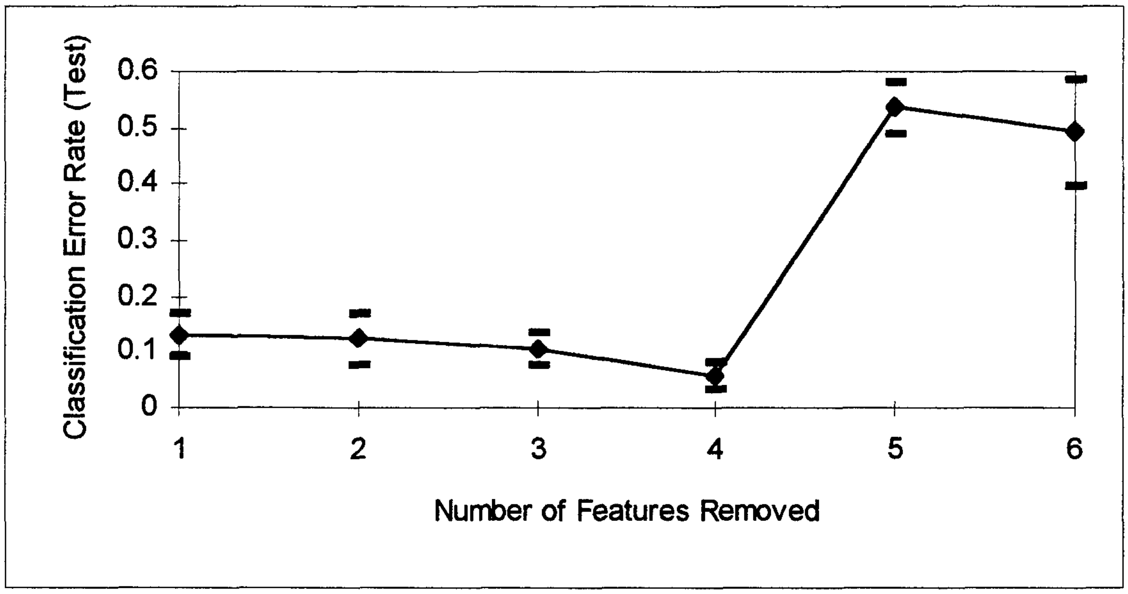

Figure 23. Average Classification Error Rate at Feature Removal
(Design Point 12 - 0.3 - 0.7, 10 Replications)

Figure 24 shows the features' average ranks. Recall that bigger is better, and since only five features are eligible for removal, the highest rank is five. Note that not only are the features ordered as expected (Noise 1, Noise 2 and Noise 3 in the first group, Variable 1 and Variable 2 in the second group) but the variability associated with the ranks of Variable 1 and Variable 2 is much smaller than that associated with the ranks of Noise 1, Noise 2 and Noise 3. The neural network has no confusion regarding the relative saliency of Variable 1 and Variable 2.

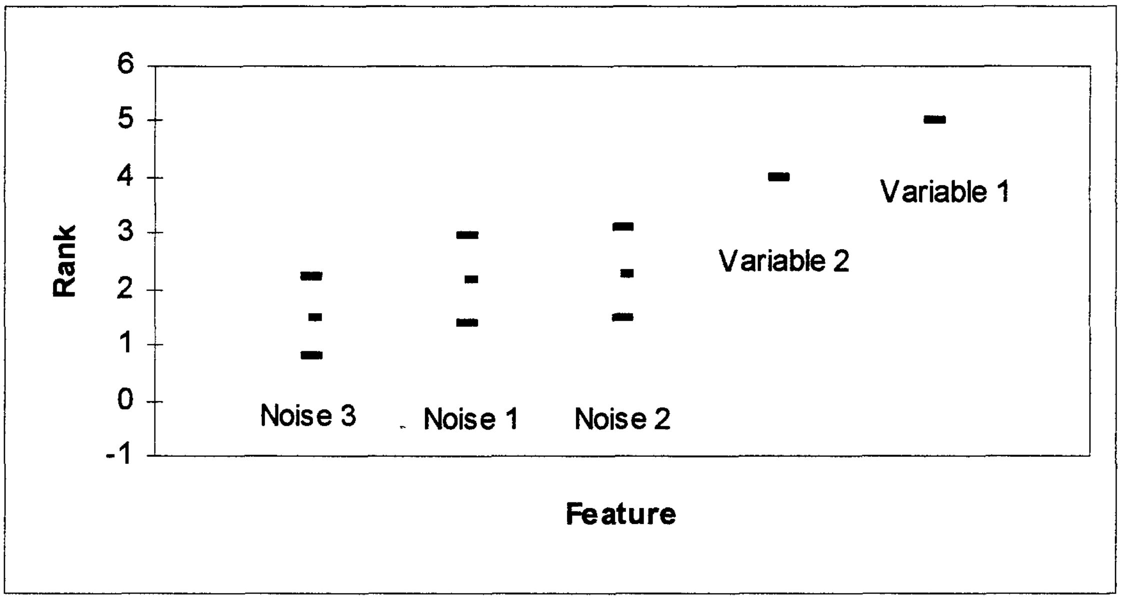

Figure 24. Average Feature Rank
(Design Point 12 - 0.3 - 0.7, 10 Replicates)

V. Final Results and Recommendations

Final Results

This thesis effort introduces a new saliency metric, the SN saliency metric, which is based upon signal-to-noise ratios, and explores its use in a new saliency screening method, the SN saliency screening method. Confidence in the SN saliency screening method depends largely upon consist feature order when ranked according to the SN saliency metric.

Chapter Three presents a designed experiment which aims at determining the consistency of the feature ranks provided by the SN saliency metric, both within and across a range of neural network architectures. Chapter Four uses the results of this designed experiment to answer four questions:

(1) Are the distribution of the signal-to-noise ratios identical for all features? Identical distributions across the features suggest that the neural network is unable to distinguish relative saliency among the features using the SN saliency metric. In that case, feature ranks based upon the SN saliency metric are highly unlikely to be consistent.

For both example problems, however, both the Kruskal-Wallis H-Test and histograms of each feature's rank suggest that the distributions are not identical. Due to violation of the independence assumption, the Kruskal-Wallis H-Test allowed no inferred conclusion and was used for illustration only.

(2) If the distributions are different, how does each feature's distribution compare to that of the injected noise feature? Salient features should generate larger SN saliency

values than nonsalient features. The injected noise feature, Noise, is a known nonsalient feature used as a reference against which to assess the saliency of the remaining features. Salient features should produce larger SN saliency values than those produced by Noise, so the SN saliency metric distribution should be shifted to the right of the same distribution for Noise. An upper-tailed Paired *t*-Test identifies features whose distributions are shifted to the right.

For the XOR problem, Variable 1 and Variable 2 are consistently identified by the Paired *t*-Test, while for the Fischer Iris Classification problem, Variable 2, Variable 3 and Variable 4 are consistently identified, while Noise 2 is occasionally identified. These results suggest that the identified features are salient.

(3) Are the feature ranks obtained from the signal-to-noise ratios consistent? If the feature ranks are not consistent, then the results of the SN saliency screening method are not reliable. Spearman's Rank Correlation Tests suggest that the feature ranks are consistent.

(4) Does network architecture affect these results? The results of the Kruskal-Wallis H-Tests, Paired *t*-Tests, and Spearman's Rank Correlation Tests suggest that a high momentum rate (0.9) might adversely affect feature relative saliency assessment, which might adversely affect the reliability of the SN saliency screening method. The test results suggest this more strongly for the Fischer Iris Classification problem than for the XOR problem.

Furthermore, graphical analysis of the effect of neural network architecture on rank distribution supports this suggestion; once again, the graphs suggest this more strongly for the Fischer Iris Classification problem than for the XOR problem. Either way, neural networks with high momentum rates might be avoided when using the SN saliency screening method.

After answering the preceding four questions, several SN saliency screenings are accomplished to investigate this method's robustness, both across multiple screenings with a given neural network and across single screenings using different neural networks. These screenings produced consistent results within both the Fischer Iris Classification problem, after accounting for redundant features, and the XOR problem.

Overall, the SN saliency screening method seems to reliably select salient features from the data sets used with the two example problems. It seems fairly robust to the different neural network architectures included in the design region. The method shows promise as a rough saliency screening tool, allowing the user to reduce the size of a data set, in one or a small number of runs, in anticipation of further screening using a finer tool. While this effort establishes no statistically validity for this method, it suggests empirical validity, warranting further research and development.

Recommendations

Recommended areas of further study include the following:

(1) Direct comparison of the SN Saliency Screening method to both the Belue-Bauer and Steppe Bauer Saliency Screening methods;

(2) Use of the SN Saliency metric in both the Belue-Bauer and Steppe-Bauer Screening methods;

(3) Statistical validation of the SN Saliency Screening method;

(4) Application of the SN Saliency Screening method to real-world problems.

Appendix A: Fisher Iris Classification Data

Table 17. Fisher Iris Classification Data.

Noise	Var 1	Var 2	Var 3	Var 4	Noise 1	Noise 2	Noise 3	Noise 4	Group 1	Group 2	Group 3
0.022045	50	33	14	2	63	24	37	15	0	0	1
0.381408	64	28	56	22	55	30	46	2	1	0	0
0.3454	65	28	46	15	55	25	49	20	0	1	0
0.19256	67	31	56	24	51	29	13	14	1	0	0
0.15263	63	28	51	15	48	30	45	2	1	0	0
0.202112	46	34	14	3	55	32	19	23	0	0	1
0.245901	69	31	51	23	55	30	42	10	1	0	0
0.929237	62	22	45	15	57	30	17	4	0	1	0
0.019012	59	32	48	18	52	27	52	18	0	1	0
0.417046	46	36	10	2	59	30	39	21	0	0	1
0.056329	61	30	46	14	52	35	59	15	0	1	0
0.080322	60	27	51	16	65	35	41	2	0	1	0
0.886047	65	30	52	20	50	36	52	1	1	0	0
0.16553	56	25	39	11	49	36	16	2	0	1	0
0.923119	65	30	55	18	65	25	45	16	1	0	0
0.029805	58	27	51	19	63	31	40	15	1	0	0
0.30158	68	32	59	23	59	25	44	23	1	0	0
0.384288	51	33	17	5	57	26	61	11	0	0	1
0.648017	57	28	45	13	52	24	39	20	0	1	0
0.396847	62	34	54	23	56	27	13	2	1	0	0
0.868973	77	38	67	22	58	37	17	15	1	0	0
0.963004	63	33	47	16	71	32	40	13	0	1	0
0.55725	67	33	57	25	61	30	47	2	1	0	0
0.447546	76	30	66	21	58	34	40	3	1	0	0
0.900543	49	25	45	17	64	30	43	23	1	0	0
0.460105	55	35	13	2	44	30	48	17	0	0	1
0.229432	67	30	52	23	67	25	50	16	1	0	0
0.370636	70	32	47	14	67	28	15	23	0	1	0
0.969775	64	32	45	15	52	30	40	13	0	1	0
0.966679	61	28	40	13	69	35	47	24	0	1	0
0.668441	48	31	16	2	50	27	15	18	0	0	1
0.411206	59	30	51	18	58	28	55	20	1	0	0
0.554037	55	24	38	11	47	36	15	2	0	1	0
0.228525	63	25	50	19	54	22	61	23	1	0	0
0.197843	64	32	53	23	57	27	13	12	1	0	0
0.893289	52	34	14	2	63	29	49	2	0	0	1
0.217766	49	36	14	1	69	36	51	23	0	0	1
0.752164	54	30	45	15	63	31	14	4	0	1	0
0.390443	79	38	64	20	46	33	45	4	1	0	0
0.11019	44	32	13	2	67	30	51	1	0	0	1
0.799705	67	33	57	21	50	25	58	13	1	0	0

Table 17 (Continued). Fisher Iris Classification Data

Noise	Var 1	Var 2	Var 3	Var 4	Noise 1	Noise 2	Noise 3	Noise 4	Group 1	Group 2	Group 3
0.41794	50	35	16	6	70	28	47	23	0	0	1
0.012878	44	30	13	2	51	26	56	11	0	0	1
0.550127	77	28	67	20	56	32	14	2	1	0	0
0.190646	63	27	49	18	68	28	17	15	1	0	0
0.272679	47	32	16	2	60	26	15	2	0	0	1
0.989417	55	26	44	12	57	31	15	13	0	1	0
0.218794	50	23	33	10	64	35	58	2	0	1	0
0.757459	72	32	60	18	67	29	54	3	1	0	0
0.391665	48	30	14	3	55	34	50	12	0	0	1
0.964584	51	38	16	2	46	32	42	2	0	0	1
0.459093	61	30	49	18	58	40	15	24	1	0	0
0.872543	48	34	19	2	46	30	49	19	0	0	1
0.344737	50	30	16	2	50	31	14	2	0	0	1
0.5635	50	32	12	2	50	26	39	19	0	0	1
0.372503	61	26	56	14	50	30	48	4	1	0	0
0.483572	64	28	56	21	61	41	35	18	1	0	0
0.966834	43	30	11	1	54	23	45	2	0	0	1
0.333953	58	40	12	2	62	35	15	15	0	0	1
0.416162	51	38	19	4	58	32	51	18	0	0	1
0.786499	67	31	44	14	63	38	61	15	0	1	0
0.643883	62	28	48	18	43	32	41	22	1	0	0
0.889974	49	30	14	2	63	37	44	18	0	0	1
0.620373	51	35	14	2	69	32	10	13	0	0	1
0.944141	56	30	45	15	51	28	15	21	0	1	0
0.584842	58	27	41	10	60	37	47	2	0	1	0
0.899539	50	34	16	4	77	31	50	3	0	0	1
0.701322	46	32	14	2	47	29	16	15	0	0	1
0.594694	60	29	45	15	62	27	33	10	0	1	0
0.804132	57	26	35	10	57	31	51	18	0	1	0
0.637877	57	44	15	4	61	30	60	24	0	0	1
0.742738	50	36	14	2	65	30	15	10	0	0	1
0.240559	77	30	61	23	64	22	49	10	1	0	0
0.52612	63	34	56	24	51	28	55	3	1	0	0
0.218645	58	27	51	19	60	34	66	12	1	0	0
0.830161	57	29	42	13	49	28	14	14	0	1	0
0.53005	72	30	58	16	57	27	63	1	1	0	0
0.432066	54	34	15	4	64	29	64	2	0	0	1
0.4736	52	41	15	1	67	28	57	10	0	0	1
0.888938	71	30	59	21	66	28	60	1	1	0	0
0.35543	64	31	55	18	77	33	14	13	1	0	0
0.970588	60	30	48	18	50	25	56	3	1	0	0

Table 17 (Continued). Fisher Iris Classification Data

Noise	Var 1	Var 2	Var 3	Var 4	Noise 1	Noise 2	Noise 3	Noise 4	Group 1	Group 2	Group 3
0.907215	63	29	56	18	72	28	13	14	1	0	0
0.539629	49	24	33	10	68	39	44	23	0	1	0
0.743644	56	27	42	13	55	38	15	12	0	1	0
0.448921	55	42	14	2	57	35	15	2	0	0	1
0.066035	57	30	42	12	49	27	16	18	0	1	0
0.30239	77	26	69	23	61	30	14	3	1	0	0
0.033611	60	22	50	15	46	34	35	13	1	0	0
0.544835	54	39	17	4	58	30	47	21	0	0	1
0.316499	66	29	46	13	79	25	41	1	0	1	0
0.608095	52	27	39	14	67	32	14	2	0	1	0
0.306119	60	34	45	16	61	38	56	15	0	1	0
0.28664	50	34	15	2	65	31	46	14	0	0	1
0.798603	44	29	14	2	53	32	15	21	0	0	1
0.929802	50	20	35	10	56	39	17	2	0	1	0
0.724236	55	24	37	10	49	29	53	19	0	1	0
0.584209	58	27	39	12	64	34	67	2	0	1	0
0.653943	47	32	13	2	67	33	14	10	0	0	1
0.849551	46	31	15	2	72	34	16	16	0	0	1
0.13162	69	32	57	23	65	28	16	22	1	0	0
0.867521	62	29	43	13	50	38	14	16	0	1	0
0.842547	74	28	61	19	48	30	67	4	1	0	0
0.567013	59	30	42	15	77	31	59	2	0	1	0
0.647119	51	34	15	2	51	27	14	15	0	0	1
0.857391	50	35	13	3	50	30	13	13	0	0	1
0.342801	56	28	49	20	66	30	56	4	1	0	0
0.505257	60	22	40	10	60	44	38	13	0	1	0
0.437562	73	29	63	18	54	34	69	14	1	0	0
0.622567	67	25	58	18	49	34	51	14	1	0	0
0.137502	49	31	15	1	54	34	43	3	0	0	1
0.61439	67	31	47	15	74	29	56	25	0	1	0
0.764627	63	23	44	13	64	30	45	18	0	1	0
0.822172	54	37	15	2	48	26	12	5	0	0	1
0.404298	56	30	41	13	68	34	45	6	0	1	0
0.60006	63	25	49	15	56	20	12	21	0	1	0
0.186782	61	28	47	12	64	23	42	18	0	1	0
0.554727	64	29	43	13	49	23	30	25	0	1	0
0.574824	51	25	30	11	51	31	57	12	0	1	0
0.261008	57	28	41	13	61	30	13	2	0	1	0
0.157423	65	30	58	22	51	33	14	13	1	0	0
0.983194	69	31	54	21	69	38	19	19	1	0	0
0.677505	54	39	13	4	62	29	48	18	0	0	1
0.896494	51	35	14	3	72	32	13	20	0	0	1
0.691955	72	36	61	25	67	28	51	20	1	0	0

Table 17 (Continued). Fisher Iris Classification Data

Noise	Var 1	Var 2	Var 3	Var 4	Noise 1	Noise 2	Noise 3	Noise 4	Group 1	Group 2	Group 3
0.110934	65	32	51	20	63	30	45	4	1	0	0
0.431005	61	29	47	14	48	34	54	21	0	1	0
0.726018	56	29	36	13	56	23	11	13	0	1	0
0.45133	69	31	49	15	59	32	44	2	0	1	0
0.336345	64	27	53	19	60	29	14	14	1	0	0
0.571441	68	30	55	21	60	33	50	2	1	0	0
0.673593	55	25	40	13	57	31	48	25	0	1	0
0.893948	48	34	16	2	51	31	14	18	0	0	1
0.192514	48	30	14	1	73	28	40	2	0	0	1
0.663474	45	23	13	3	56	30	51	22	0	0	1
0.71871	57	25	50	20	48	34	49	15	1	0	0
0.880614	57	38	17	3	50	30	33	15	0	0	1
0.763229	51	38	15	3	54	25	16	14	0	0	1
0.512937	55	23	40	13	45	27	53	11	0	1	0
0.127309	66	30	44	14	54	30	51	19	0	1	0
0.30831	68	28	48	14	62	30	58	18	0	1	0
0.020578	54	34	17	2	58	22	42	13	0	0	1
0.67425	51	37	15	4	51	32	16	20	0	0	1
0.40357	52	35	15	2	44	42	56	17	0	0	1
0.881294	58	28	51	24	76	29	46	2	1	0	0
0.723314	67	30	50	17	77	38	15	13	0	1	0
0.06308	63	33	60	25	55	32	45	2	1	0	0
0.488632	53	37	15	2	63	33	55	2	0	0	1

Appendix B: XOR Classification Data

Table 18. XOR Classification Data

Noise	Variable 1	Variable 2	Noise 1	Noise 2	Noise 3	Group 1	Group 2
-0.3207	-0.3438	0.3951	0.3716	0.0476	0.0908	0	1
-0.4727	0.4342	-0.2489	-0.0131	-0.069	-0.8866	0	1
0.1647	-0.6473	0.0224	0.4967	0.618	-0.9706	0	1
0.2865	-0.3052	0.23	-0.9012	0.4726	-0.6751	0	1
0.3985	0.7185	-0.633	-0.4905	-0.4538	0.3337	0	1
0.5439	0.0181	-0.5299	0.3239	-0.5213	0.0216	0	1
0.1544	0.8304	-0.7728	0.7054	0.6681	-0.2308	0	1
-0.181	0.4922	-0.771	0.7707	-0.9853	0.4334	0	1
-0.93	0.7999	-0.2749	0.5235	0.1763	0.4262	0	1
0.5665	0.0362	-0.1976	0.0363	0.5994	0.1086	0	1
-0.4034	-0.65	0.9569	-0.3082	-0.3947	0.5913	0	1
0.5414	-0.8746	0.0332	0.7607	0.8394	-0.9111	0	1
-0.8526	0.3035	-0.5517	0.3742	-0.2504	0.9364	0	1
0.3549	-0.4809	0.7476	0.2692	0.9663	-0.7978	0	1
-0.0319	-0.5848	0.371	-0.018	0.6774	0.7371	0	1
-0.3009	0.4845	-0.5842	-0.9161	-0.5019	0.8058	0	1
-0.3012	0.4994	-0.8499	-0.3154	-0.4209	0.9874	0	1
0.0991	-0.9364	0.7694	-0.8905	-0.4757	0.2223	0	1
-0.6187	-0.6927	0.5857	0.6848	0.2447	-0.9645	0	1
0.5982	-0.0512	0.0906	0.27	-0.6106	-0.1088	0	1
0.2753	0.7003	-0.9421	-0.0364	0.7919	0.313	0	1
-0.7117	0.014	-0.3179	0.0849	0.0798	-0.4482	0	1
-0.7124	0.1642	-0.1129	-0.7806	-0.3442	-0.7788	0	1
0.4145	-0.3602	0.3625	0.728	-0.1739	0.9005	0	1
-0.3615	0.5011	-0.206	-0.6123	0.9687	-0.1116	0	1
0.0434	0.6386	-0.5731	-0.4276	-0.3779	-0.1033	0	1
0.5688	-0.4039	0.8917	-0.5468	0.8999	0.1988	0	1
-0.5865	0.0686	-0.0841	0.2135	-0.6587	0.6161	0	1
0.6612	0.6533	-0.0234	0.7759	-0.5144	0.7753	0	1
0.937	-0.5806	0.9703	-0.5317	-0.3151	0.2479	0	1
0.7585	-0.8137	0.4918	0.0211	-0.4356	-0.9853	0	1
0.3347	0.3475	-0.5494	-0.889	0.0797	0.7607	0	1
-0.8357	0.789	-0.534	0.485	0.8462	-0.7891	0	1
-0.3361	-0.5197	0.1475	-0.7969	0.3144	0.631	0	1
-0.4056	-0.5044	0.1056	-0.2702	0.6518	0.3993	0	1

Table 18 (Continued). XOR Classification Data

0.6133	-0.2088	0.7834	0.6078	0.2078	-0.1937	0	1
0.0989	0.0395	-0.4349	-0.7286	-0.795	0.7811	0	1
-0.3076	-0.5126	0.7378	0.3904	0.2954	0.0999	0	1
-0.5895	-0.3628	0.4459	-0.6382	-0.7452	-0.7733	0	1
-0.871	-0.22	0.1245	0.6976	0.058	-0.3287	0	1
0.1432	0.299	-0.1477	-0.1777	-0.4409	0.6587	0	1
0.672	-0.7658	0.7635	0.3921	-0.1046	-0.6544	0	1
0.1889	0.7548	-0.0294	-0.427	0.5141	-0.7239	0	1
-0.6156	-0.9245	0.9954	0.347	-0.953	0.6112	0	1
0.7215	-0.7919	0.3899	-0.2433	0.9099	0.6846	0	1
0.2055	-0.0699	0.7829	0.9385	0.1917	-0.5503	0	1
0.3733	-0.1733	0.9466	0.7269	-0.5664	0.2741	0	1
0.9483	0.4314	-0.3319	0.8096	-0.1985	-0.6921	0	1
-0.4764	0.8225	-0.1494	-0.4578	-0.7725	0.5042	0	1
-0.2706	0.2043	-0.9815	0.7128	-0.247	0.8288	0	1
0.523	-0.3248	0.8413	0.63	0.3467	0.4133	0	1
-0.0888	0.0028	-0.6916	0.2368	-0.1889	0.3619	0	1
-0.0923	0.6141	-0.2804	0.8481	-0.8132	-0.393	0	1
-0.3508	0.7995	-0.3425	0.6666	-0.4619	0.5378	0	1
-0.8783	0.149	-0.8412	0.591	-0.3809	-0.2529	0	1
-0.9681	-0.0137	0.2119	-0.4643	0.0939	0.3053	0	1
-0.6519	-0.7495	0.9102	-0.0176	-0.3332	-0.827	0	1
0.0627	0.4931	-0.8917	-0.8833	-0.3101	-0.8444	0	1
0.0651	0.375	-0.4276	0.9831	0.9707	-0.3431	0	1
-0.2927	0.7211	-0.3766	-0.5202	-0.4024	0.401	0	1
0.207	0.9749	-0.7212	-0.7524	0.6984	0.819	0	1
0.3593	-0.5726	0.9543	0.551	0.355	-0.5948	0	1
-0.1237	-0.7888	0.323	0.1993	-0.2713	-0.5148	0	1
-0.185	0.4899	-0.4094	-0.7546	0.6538	0.0456	0	1
-0.4652	0.1682	-0.8528	-0.502	-0.2938	0.772	0	1
-0.6674	0.2614	-0.8452	-0.0898	0.8419	-0.5924	0	1
-0.398	-0.3781	0.3439	-0.6011	0.6776	-0.762	0	1
-0.0445	0.4357	-0.8722	0.5617	-0.4356	-0.745	0	1
-0.9901	-0.404	0.9566	-0.9078	0.5602	0.9507	0	1
0.8858	-0.2806	0.8611	-0.4713	-0.0651	-0.9231	0	1
-0.5884	0.6328	-0.3075	0.677	-0.4657	-0.9833	0	1
0.5104	-0.1472	0.1586	-0.3882	-0.4879	0.4834	0	1
0.7609	0.339	-0.2293	0.5197	-0.9079	-0.1185	0	1
-0.1408	0.9286	-0.7147	-0.6844	-0.9643	-0.0604	0	1
-0.499	-0.4581	0.7671	-0.229	0.085	-0.6055	0	1

Table 18 (Continued). XOR Classification Data

0.1576	0.0451	-0.1202	-0.2276	-0.6333	-0.8513	0	1
-0.1153	-0.4004	0.7903	0.0214	0.7662	-0.2703	0	1
-0.3891	-0.6353	0.3547	0.3021	-0.5114	0.8196	0	1
-0.1823	0.1761	0.1646	-0.9225	0.1283	0.0622	1	0
0.8917	-0.4054	-0.2004	0.3323	0.1559	0.5693	1	0
-0.5389	-0.3217	-0.2696	-0.8776	-0.273	0.8677	1	0
-0.9269	-0.0926	-0.6453	0.6108	-0.3659	-0.6608	1	0
-0.4483	0.3844	0.9998	-0.305	-0.8642	-0.8744	1	0
-0.3656	0.9352	0.8664	-0.2544	0.6247	0.5672	1	0
-0.1839	0.5261	0.1536	0.891	0.0613	0.8216	1	0
-0.0531	-0.9997	-0.635	0.0741	0.7382	-0.0059	1	0
0.8289	-0.66	-0.7642	-0.352	0.3797	0.4634	1	0
-0.471	-0.5249	-0.8349	0.6632	-0.8406	-0.8778	1	0
-0.5896	-0.8409	-0.3275	-0.0953	-0.5993	-0.5169	1	0
0.9039	-0.9846	-0.8121	-0.7448	-0.9454	0.7714	1	0
0.8203	0.083	0.642	-0.7611	0.3321	-0.5577	1	0
0.7313	-0.8519	-0.6884	-0.3862	0.2806	-0.1668	1	0
0.6865	-0.0665	-0.389	0.3695	0.1109	0.87	1	0
0.1679	-0.849	-0.9097	0.6086	-0.7727	0.6331	1	0
0.8907	-0.3606	-0.3024	0.9413	-0.2811	0.6132	1	0
0.5459	-0.0974	-0.0492	0.8526	-0.5258	0.0096	1	0
-0.4808	0.5801	0.1066	-0.5855	0.2593	-0.4845	1	0
0.8791	-0.1934	-0.1573	-0.366	-0.7449	-0.688	1	0
-0.2173	-0.576	-0.3753	0.2778	0.7095	-0.9209	1	0
0.0531	0.0158	0.463	-0.6815	-0.3141	-0.7275	1	0
-0.0425	-0.6564	-0.4311	0.5233	-0.2269	0.2229	1	0
0.6127	0.9401	0.4005	-0.4685	-0.0027	-0.7518	1	0
-0.3003	0.5842	0.9718	0.7086	-0.2763	-0.5242	1	0
0.2378	-0.6246	-0.4257	-0.4178	-0.5264	0.2689	1	0
-0.2399	-0.475	-0.7777	0.2528	0.4403	-0.453	1	0
0.4069	-0.5793	-0.379	0.6031	0.353	-0.6263	1	0
0.5199	-0.8059	-0.3021	-0.2872	-0.399	-0.0586	1	0
0.7994	-0.6771	-0.4305	0.5268	0.2745	0.8132	1	0
-0.9327	0.9171	0.4205	0.2317	0.0076	-0.5235	1	0
0.355	0.2076	0.497	-0.9196	0.815	-0.9283	1	0
0.6401	-0.8852	-0.6779	0.6242	0.2856	0.5286	1	0
-0.2965	-0.4585	-0.4532	-0.4408	0.1266	-0.6969	1	0
-0.7495	-0.919	-0.1059	0.6452	-0.8293	-0.8899	1	0
-0.7618	0.833	0.4735	0.5941	-0.4412	-0.3377	1	0
-0.9926	0.1994	0.891	0.4922	0.9791	0.7353	1	0

Table 18 (Continued). XOR Classification Data

-0.5399	-0.1275	-0.2263	0.2757	0.6092	-0.2447	1	0
-0.1276	0.9356	0.5549	0.2313	-0.28	0.5784	1	0
-0.911	0.8981	0.0235	0.7904	0.8437	0.4684	1	0
-0.2695	0.1781	0.7904	0.8606	0.8513	-0.2233	1	0
-0.6636	0.0731	0.3571	-0.4533	-0.2875	0.4773	1	0
0.9935	0.1532	0.5018	0.8172	0.8974	-0.7721	1	0
0.8922	-0.1754	-0.6651	0.7649	0.0553	-0.454	1	0
-0.0617	-0.0256	-0.3691	-0.6798	-0.9585	-0.8156	1	0
-0.9229	0.8401	0.1388	-0.5639	0.6544	0.921	1	0
-0.7383	0.3322	0.0534	0.8452	-0.7723	-0.5729	1	0
0.9536	-0.4805	-0.1914	-0.9499	-0.4389	-0.1251	1	0
0.9631	0.4717	0.8687	0.2782	-0.7399	0.7264	1	0
0.6798	0.4337	0.5954	0.0432	0.8878	-0.2678	1	0
0.9298	0.3394	0.6821	-0.1907	-0.9678	-0.5892	1	0
-0.0123	-0.5569	-0.8104	-0.2052	0.0342	-0.7574	1	0
0.0209	-0.0165	-0.9735	-0.5543	0.4767	0.1343	1	0
0.2392	-0.1024	-0.2209	-0.8122	0.6845	0.2924	1	0
-0.9763	-0.5912	-0.5132	0.4391	-0.4879	0.306	1	0
-0.9494	-0.8562	-0.6583	-0.8846	0.5606	-0.3017	1	0
-0.2336	0.4334	0.1204	0.676	-0.5054	0.7048	1	0
-0.1623	0.6303	0.3886	0.4157	0.7527	0.8091	1	0
0.6854	0.6488	0.8034	0.5925	0.7065	0.4623	1	0
0.3339	0.0242	0.2018	0.1503	-0.8691	0.5174	1	0
-0.4653	-0.3967	-0.4882	-0.2609	-0.8796	0.1616	1	0
-0.8214	-0.4844	-0.3983	-0.8921	0.9582	-0.4049	1	0
0.9801	0.0851	0.6796	0.9963	0.8416	-0.2775	1	0
-0.3369	0.9954	0.659	0.1454	-0.4061	0.0849	1	0
-0.4738	0.2006	0.5809	-0.8448	-0.3217	0.6124	1	0
-0.6458	0.3577	0.2922	-0.2822	-0.3738	-0.5482	1	0
-0.9204	-0.279	-0.3688	-0.0029	-0.0827	0.1813	1	0
-0.3438	-0.998	-0.2363	-0.1231	0.1435	0.6049	1	0
0.3477	-0.7284	-0.4753	0.0994	-0.2503	-0.9569	1	0
-0.5216	0.9481	0.7547	-0.1315	-0.2637	0.0042	1	0
-0.357	-0.6946	-0.5994	-0.5846	0.5808	0.8003	1	0
0.5131	0.2127	0.5883	-0.1298	0.217	-0.2127	1	0

Bibliography

1. Belue, Lisa M. An Investigation of Multilayer Perceptrons for Classification. MS thesis, AFIT/GOR/ENS/92M-02. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1992.

2. Belue, Lisa M. and Kenneth W. Bauer, Jr. "Determining Input Features for Multilayer Perceptrons," Neurocomputing, 7: 111-121 (1995).

3. Defense Advanced Research Projects Agency (DARPA). Neural Network Study AFCEA International Press, Fairfax VI, November 1988.

4. Hines, William W. and Douglas C. Montgomery. Probability and Statistics in Engineering and Management Science (Third Edition) . New York: John Wiley & Sons, 1990.

5. Mendenhall, William and others. Mathematical Statistics with Applications (Fourth Edition). Belmont, California: Duxbury Press, 1990.

6. Neter, John and others. Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Designs (Third Edition). Burr Ridge, Illinois: Richard D. Irwin, 1990.

7. Reinhart, Gregory L. A FORTRAN Based Learning System Using Multilayer Back-Propagation Neural Network Techniques. MS thesis, AFIT/GOR/ENS/94M-11. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1994.

8. Ruck, Capt Dennis W. Characterization of Multilayer Perceptrons and Their Application to Multisensor Automatic Target Detection. Phd dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1990 (AD-A229035).

9. Steppe, Jean M. Feature and Model Selection in Feedforward Neural Networks. PhD dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1994.

10. Tarr, Capt Gregory L. Multi-layered Feedforward Neural Networks for Image Segmentation. PhD dissertation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, November 1991.

Vita

Captain David B. Sumrell [redacted], into a military family. He graduated from Coronado High School, Colorado Springs, Colorado, in 1981, and enjoyed a short summer before taking the short drive to the United States Air Force Academy to begin his military career. He graduated in 1985 with a Bachelor of Science Degree. He was selected for Undergraduate Pilot Training and, upon completion, was assigned as a B-52 pilot stationed at Fairchild AFB, Washington. Shortly after upgrading to aircraft commander in 1990, AETC invited Captain Sumrell to return to the T-37 as a UPT instructor pilot at Reese AFB, Texas. Four rewarding years later, Captain Sumrell received an invitation to pursue a Masters Degree in Operations Research at the Air Force Institute of Technology, which he accepted. His academic odyssey began upon his arrival in August of 1995.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1996

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
AN INVESTIGATION OF PRELIMINARY FEATURE SCREENING USING SIGNAL-TO-NOISE RATIOS

5. FUNDING NUMBERS

6. AUTHOR(S)
David B. Sumrell, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/96M-17

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A new saliency metric and a new saliency screening method are developed. This new metric, the SN saliency metric, is based upon signal-to-noise ratios, where the signal is provided by a sum of squared weights associated with a given feature, and the noise is based upon a sum of squared weights associated with a reference noise feature which is injected into the data. The resultant metric allows for a direct comparison of the feature of interest with a reference noise feature which is known to be nonsalient. The SN saliency screening method, which uses the SN saliency metric, offers the potential of identifying salient features in one saliency screening run and is envisioned as an economical rough screening tool to be used prior to more refined screening efforts or more exhaustive training efforts. During the screening run, features are removed individually based upon their rank as determined by the SN saliency metric. The classification error rate's reaction to a given feature's removal helps confirm that feature's saliency.

14. SUBJECT TERMS
Multilayer perceptron; Feature selection; Saliency

15. NUMBER OF PAGES
90

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL